Platform Diffusion at Temporary Gatherings: Social Coordination and Ecosystem Emergence

Accepted at Strategic Management Journal, *6 August 2020*

Published online at https://doi.org/10.1002/smj.3230

Tommy Pan Fang
PhD Candidate in Business Administration
Technology and Operations Management Unit
Harvard Business School
Harvard University
ORCID: 0000-0003-2332-7121
Address: 60 Harvard Way, Cotting House 302C, Boston, MA 02163 USA
Email: tpanfang@hbs.edu

Andy Wu*
Assistant Professor of Business Administration
Strategy Unit
Harvard Business School
Harvard University
ORCID: 0000-0002-9107-5731
Address: 15 Harvard Way, Morgan Hall 243, Boston, MA 02163-1011 USA
Email: awu@hbs.edu

David R. Clough
Assistant Professor
Organizational Behaviour and Human Resources Division
Entrepreneurship and Innovation Group
Sauder School of Business
University of British Columbia
ORCID: 0000-0002-2556-7033
Address: 2053 Main Mall, Vancouver, BC V6T 1Z2 Canada
Email: david.clough@sauder.ubc.ca

Running Head: Platform Diffusion at Temporary Gatherings
Keywords: Innovation Ecosystems, Multi-Sided Platforms, Technology Diffusion, Hackathons, Complex Contagions

*Corresponding author.

Platform Diffusion at Temporary Gatherings:
Social Coordination and Ecosystem Emergence

ABSTRACT

Research Summary

Software platforms create value by cultivating an ecosystem of complementary products and services. Existing explanations for how a prospective complementor chooses platforms to join assume the complementor has rich information about the range of available platforms. However, complementors lack this information in many ecosystems, raising the question of how complementors learn about platforms in the first place. We investigate whether attending a temporary gathering—a hackathon—impacts the platform choices of software developers. Through a large-scale quantitative study of 1,302 developers and 167 hackathons, supported by qualitative research, we analyze the multiple channels—sponsorship, social learning, knowledge exchange, and social coordination—through which hackathons serve as a social forum for the diffusion of platform adoption among attendees.

Managerial Summary

A software platform such as Windows, iOS, or Amazon Web Services relies on third-party developers to create applications that complement the platform and make it valuable for end users. However, developers face a wide range of possible platforms, and they may have limited information about which platforms would be worthwhile to develop for. A software platform business can educate and encourage developers to adopt their platform by supporting in-person software development competitions, known as hackathons. Developers learn about prospective platforms that advertise at the hackathon. Developers also learn whether and how to use a platform by observing and teaching one other. Hackathons are particularly useful for spreading platform technologies: developers prefer to adopt widely-used platforms, and hackathons permit developers to identify and join fashionable platforms.

INTRODUCTION

Software platforms create value for their users by cultivating an ecosystem of complementary products and services (Brandenburger & Nalebuff, 1996; Cennamo & Santalo, 2013; Jacobides, Cennamo, & Gawer, 2018; Kapoor, 2018; Tiwana, 2015). Scholars of platform ecosystems therefore study how complementors select one or more platforms to join (Boudreau, 2010; Church & Gandal, 1992; Gawer & Henderson, 2007; Zhu & Iansiti, 2012).

Studies of complementor decision-making often assume the complementor possesses rich information on the platforms they are choosing between, e.g., information about the user base, competitive intensity, and governance rules. While that assumption may hold in certain well-established ecosystems, in other settings complementors lack the rich information they would need to make a calculative choice between alternative platforms (Dattée, Alexy, & Autio, 2018). In emerging or fast-changing ecosystems, the set of platform players may be ambiguous and data on platform usage may be unreliable or non-existent (Dougherty & Dunne, 2011; Hannah & Eisenhardt, 2018; Wade, 1995). In the ecosystem of general software programming—in which software developers are complementors—the domain of possible platforms a developer can use is huge and the range of possible applications they could develop is vast (Eisenmann, 2006; Kapoor & Agarwal, 2017). Thus, platform research must address the question of how complementors learn about platforms in the first place.

In fast-changing ecosystems, technology platform companies proactively market themselves to prospective complementors (Cusumano & Gawer, 2002). These companies sometimes use in-person events to attract complementors and to motivate them to build products that complement the platform (West & Wood, 2013; Yoffie, Casadesus-Masanell, & Mattu, 2003). Although prior research has noted the existence of these events, and flagged their possible influence on complementor behavior (e.g., Dattée et al., 2018; Özalp, Cennamo, & Gawer, 2018), the impact of temporary gatherings on platform growth has not yet been studied in detail. In this paper we undertake such a study. We use the term *temporary gatherings* to encompass formally organized in-person events lasting between several hours and several days.[1] We address the questions: How does attendance at a temporary gathering affect the platform adoption decision of a complementor? And can platform owners cultivate an ecosystem of complementors by sponsoring these events?

The impact of temporary gatherings on platform adoption is theoretically interesting for two main reasons. First, although theories of technology diffusion emphasize social interactions as the basis of information flows, it is unclear whether transient interactions between strangers influence their subsequent platform adoption choices. Prior work suggests strong network ties are the most influential in the choice of a technology (Suarez, 2005; Venkatraman & Lee, 2004). Multiple, persistent sources of influence may be needed before people change their behavior by adopting a new technology (Centola, 2018). However, encounters at temporary gatherings tend to be transitory, weak-tie interactions; as such, they may not leave much long-term impact on an attendee (e.g., Ingram & Morris, 2007).

Second, if temporary gatherings do impact complementor behavior, then they may allow technology to diffuse between the local clusters that make up a large-scale social network (Granovetter, 1973; Gulati, Sytch, & Tatarynowicz, 2012; Watts & Strogatz, 1999). Prior research has found that technology use tends to be homogeneous within a social network cluster, with limited diffusion taking place between clusters (Lee, Lee, & Lee, 2006). For example, in a classic study of contraception techniques, Rogers & Kincaid (1981) found that most women within a given village use the same form of contraception, but between villages the method chosen varied. Technology diffusion has been characterized as a "complex contagion," defined

[1] Besides hackathons, other temporary gatherings include conferences, music festivals, art fairs, and trade shows.

as a behavior that only spreads when prospective adopters are exposed to multiple sources of social influence (Centola & Macy, 2007). Because digital platforms exhibit network effects (Church & Gandal, 1992), they are particularly likely to be subject to such threshold-based adoption dynamics (Granovetter, 1978). Current diffusion research takes a strong interest in identifying interaction structures that spread complex contagions (Assenova, 2018; Beaman et al., 2018). A distinctive attribute of temporary gatherings is that they bring people from different social clusters together. If attendees do influence one another, then temporary gatherings might be a key mechanism by which platform adoption spreads widely through a population of prospective adopters.

We focus on temporary gatherings in the form of software development hackathons, which we further describe later in the paper. In short, hackathons bring developers together to create new software applications in a short time frame (Lifshitz-Assaf, Lebovitz, & Zalmanson, 2020).[2] Software developers come to these events to learn new skills, create new applications (with teammates), and compete for prizes. Importantly for platform owners, these software developers are prospective complementors in the ecosystems of digital platforms (Eisenmann, 2006). As sponsors of the hackathon, platform owners provide financial, in-kind, and in-person logistical support to attending developers. Platform owners use these events as an opportunity to promote their platform to third-party software developers (Parker, Van Alstyne, & Jiang, 2017).

Theoretically, a hackathon serves as a social context in which attendees observe and influence one another, creating the opportunity for attendees to form network ties with each other and with sponsoring platforms (Dahlander & McFarland, 2013; Feld, 1981). Building from existing research on technology diffusion, we argue that temporary gatherings facilitate social

[2] Although 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; Flores et al., 2019; Frolund, Murray, & Riedel, 2018).

learning (Wade, 1995), knowledge exchange (Garud, 2008), and social coordination (Farrell & Klemperer, 2007; Teece, 2007). Social learning refers to the tendency of people to draw inferences from the behavior of others. Knowledge exchange refers to the active sharing of experience on how to adopt and use a technology. Going over and above these established mechanisms, we advance a theory of social coordination, which we define as decision-making based on the alignment of collective expectations through face-to-face interaction (Halaburda & Yehezkel, 2019; Schelling, 1978). Temporary gatherings, we argue, align the expectations of attendees by allowing them to identify and join emerging bandwagons (Abrahamson & Rosenkopf, 1997). Temporary gatherings also allow sentiment towards a platform to converge (Rawlings & Friedkin, 2017).

To test our theoretical framework, 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 through November 2017 as they each participate in a hackathon, taking advantage of variation in the hackathon dates and platform sponsors to study hackathons' impact on developers' platform adoption behavior.

We find quantitative and qualitative evidence to support four distinct channels through which temporary gatherings affect platform diffusion. First, we find a baseline impact of hackathon sponsorship on platform adoption by hackathon attendees, consistent with such sponsorship driving awareness and acting as an economic inducement to adopt the platform. After developers attend a hackathon sponsored by a given platform, their annual hazard of adopting the platform's Application Programming Interface (API) in their software development projects rises by 20.4 percentage points over the baseline hazard. Second, developers engage in social learning at hackathons: every 10 percentage point increment in the proportion of attendees who are prior platform adopters is positively associated with a 1.2 percentage point increase in a developer’s annual hazard of adopting the platform. Third, hackathon attendees actively exchange knowledge on how to adopt and use platform technologies. Fourth, hackathons are forums for social coordination. We find that platform adoption hazard rises non-linearly with the proportion of attendees who are prior platform adopters; the effect is stronger when the proportion of other adopters is high, suggesting that temporary gatherings help attendees identify platforms that are likely to be dominant. We further find that when a platform sponsors a hackathon, the association between adoption hazard and proportion of attendees who are prior adopters is amplified, suggesting that hackathon sponsorship helps the platform foster positive sentiment.

Our study makes two contributions to the strategy literature. First, we contribute to the literature on platform ecosystems by studying the role of temporary gatherings as settings where platform owners can attract complementors (Adner, 2017; McIntyre & Srinivasan, 2017). We provide evidence that mechanisms of frequency-based social learning, outcome-based social learning, and knowledge exchange all take place at temporary gatherings, and we develop theory on how in-person, group interactions enable social coordination around a nascent technology. Second, we contribute to the literature on how social structure influences technology diffusion (Centola, 2018; Granovetter, 1978; Rogers, 2003). We identify temporary gatherings as an important context for the diffusion of platform technologies. Our findings have broad implications for how complex contagions spread through populations of prospective adopters.

THEORY AND HYPOTHESES

Navigating a multi-platform setting in poor visibility

Much existing research on how platforms attract complementors examines settings where prospective complementors are familiar with the set of platforms they can join, such as the choice between consoles in the videogame industry (e.g., Cennamo, Özalp, & Kretschmer, 2018;

Clements & Ohashi, 2005; Schilling, 2003; Shankar & Bayus, 2003) or the choice between enterprise planning software platforms (e.g., Ceccagnoli et al., 2012; Huang et al., 2013). This body of work establishes that complementors prefer to join a platform with a larger installed base of users (Boudreau & Jeppesen, 2015), less competition (Boudreau, 2012), lower technical difficulty to adopt (Özalp, Cennamo, & Gawer, 2018), and a governance structure that permits the complementor to capture value ((Huang et al., 2013; Thomas, Autio, & Gann, 2014).

However, in many settings complementors lack rich information on the platforms they can join. Dattée et al. (2018: 467) note that some ecosystems exhibit poor visibility, defined as the absence of a "compelling ecosystem blueprint that is tangible enough to reduce uncertainty." Poor visibility is associated with generative technologies—such as software development, quadrotor drones, or 3D printing—which offer such a multitude of possible applications that neither platform nor complementor can foresee how the technology will be used (Denrell, Fang, & Winter, 2003; Zittrain, 2006).[3] Facing this lack of information, a complementor may not be familiar with the menu of possible platforms they can choose from. Information about the installed base of users and competitive intensity on the platform may be unreliable or nonexistent (Mead, 2012). In addition, a platform may opaquely hide governance rules within lengthy terms and conditions (Lemley, 2006).

In these settings, technology platform companies proactively market themselves to stimulate awareness amongst prospective platform adopters (Cusumano & Gawer, 2002). These companies sometimes cultivate their user base by evangelizing the platform at an in-person event where company representatives interact with prospective complementors, aiming to motivate

[3] A piece of software often occupies an intermediate tier in a layered modular architecture (Yoo, Henfridsson, & Lyytinen, 2010); the software must interface with multiple types of platforms (e.g. an Operating System, data sources, computing resources) and it may be written flexibly to port across multiple platforms of the same type (i.e. multi-homing across several operating systems (Venkataraman, Ceccagnoli, & Forman, 2017)).

them to build products that complement the platform. For example, companies developing a new computer operating system (OS) host developer conferences to educate complementors about the platform's capabilities (West and Wood, 2013; Yoffie, Casadesus-Masanell, & Mattu, 2004). Özalp and colleagues (2018) describe how video game console owners used the Games Developers' Conference as a forum to disseminate knowledge to complementors on how to use the console's programming environment. In less-established innovation ecosystems, in-person events may alleviate complementors' perceived uncertainty over a new technology (Dattée et al., 2018; Garud, 2008).

Our study focuses on a widespread form of temporary gathering: software development hackathons. A hackathon is an event that brings together participants who generate ideas or products over a short period of time. Past studies and reports emphasize the creative and collaborative approach to problem-solving that takes place at a hackathon (e.g., Arena et al., 2017; Lifshitz-Assaf, 2018). The tight time frame of one to several days creates sprint-like conditions. Most software development hackathons are, at least nominally, structured as a competition, with prize(s) presented at a public closing ceremony, where the winner(s) and runner(s)-up demonstrate their product to a broad audience. In this study, we focus on open hackathons for software developers with a physical venue for participants to co-locate.[4] Open hackathons invite members of the public to participate, allowing us to study platform adoption by the broader developer population.[5]

Hackathons involve three main types of stakeholders: organizers, sponsors, and

---

[4] Hackathons that take place entirely online lie outside the scope of this study. In-person events that focus on other creative pursuits, such as developing a business plan, academic article, or writing music, can be considered temporary gatherings—and are even sometimes referred to as hackathons—but these events are distinct from software development hackathons. Like hackathons, crowdsourcing competitions evaluate participants and identify "winners" at the end of the time-delimited event (Boudreau, Lacetera, & Lakhani, 2011), but in other respects crowdsourcing competitions are a distinct phenomenon from hackathons (see Online Appendix A for discussion).

[5] In contrast to open hackathons, closed company hackathons limit their attendees to members of an organization or firm, e.g., Microsoft and Facebook run internal hackathons for their own employees to develop projects, of their own choosing, for the company.

developers. First, an organizer operates the hackathon and brings in the other stakeholders. Hackathon organizers consist of industry trade groups (e.g., Python Software Foundation),[6] universities and student groups (e.g., PennApps at the University of Pennsylvania), governmental entities (e.g., U.S. General Services Administration),[7] non-profit interest groups (e.g., Roman Catholic Church) (Valdez, 2018), and for-profit firms. The organizer advertises the event to potential sponsors and developers. Second, a sponsor is a third party, usually a for-profit firm, providing financial and in-kind support for the operations of the hackathon. We focus on sponsors who operate a for-profit platform as one of their main businesses. Third, a developer participates by developing software at the hackathon; a developer is a prospective complementor for the sponsors who own platform ecosystems. Software developers participate in the hackathon and are eligible to win prize(s). They select an application idea and during the hackathon they write and debug software code to build the application. Most developers spend nearly all their waking hours working, eating, or playing at the hackathon venue. Importantly for our theorizing, developers work with their teammates at tables in close proximity to other teams, and they socialize informally throughout the hackathon which can run through the night. Online Appendix A provides further contextual background on hackathons as a form of temporary gathering and on the backgrounds and motivations of all three stakeholder types.

A software developer's attendance at a temporary gathering might influence her platform adoption, both at the event and beyond it, through multiple causal pathways. Adoption is costly because the developer needs to invest time and energy to learn how to use a new platform. The following subsections outline several mechanisms which we expect may influence the developer's adoption choices. Figure 1 provides an overview of the mechanisms we discuss.

[6] See, "About PyCon," https://us.pycon.org/2019/about/ [9 January 2020].
[7] See, "GSA customer experience hackathon 2019," https://digital.gov/event/2019/06/19/gsa-customer-experience-cx-hackathon/ [9 January 2020].

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

Platform sponsorship of software development hackathons

As an empirical baseline, we describe how platform sponsorship of a software development hackathon can generate awareness of a platform. Sponsorship of a temporary gathering serves as a form of marketing to make prospective platform adopters aware of the platform's existence. In Roger's canonical model of innovation diffusion (2003), "awareness" of an innovation is the first step in a potential adopter's decision-making process. Upon first encountering a new technology, potential adopters face uncertainty as to whether they will benefit from adopting it or not (Coleman, Katz, & Menzel, 1957; cf. Greve, 2009), though some may still experiment with it (Catalini & Tucker, 2017). Technology companies that sponsor gatherings such as hackathons can use the event to raise awareness of their platform by publicizing the platform's name at the event and often by having representatives from the sponsor present. As part of the empirical baseline, we expect that sponsoring the event would make attendees more likely to adopt the platform. Sponsors of technology-focused events can also provide an economic inducement for potential complementors to join their platform (Rochet & Tirole, 2003): hackathon sponsors offer valuable prizes to incentivize teams to adopt the company's platform in their hackathon project.

As indicated in the first row of Figure 1, these baseline mechanisms refer to the influence of an event sponsor on an event attendee. These mechanisms are present independent of any social interactions amongst attendees at the event.

Temporary gatherings as forums for social learning

The next set of mechanisms examine hackathons as forums for social learning; these mechanisms are based on the influence that event attendees have on one another, as depicted in the second row of Figure 1.

When faced with uncertainty, people often look to others for guidance as to what is appropriate, legitimate, or effective (DiMaggio & Powell, 1983). Social learning refers to the tendency of people to draw inferences from the behavior of others (Bandura, 1986). Research on both individuals and organizations finds that social learning is stronger under settings of uncertainty (e.g., Gaba & Dokko, 2016; Hogg, 2010). Because prospective adopters face uncertainty over the usefulness of a new technology, their adoption decisions are often, therefore, guided by what they observe others doing (Greve & Seidel, 2015). Prior literature on social learning suggests that some inferences are drawn from observing the frequency with which a technology is used (Haveman, 1993), while other (possibly stronger) inferences are drawn from observing whether the technology is associated with positive outcomes (Clough & Piezunka, 2020; Haunschild & Miner, 1997). We suggest that both pathways may influence the choices of complementors who observe one another while at a temporary gathering.

Social learning from frequency of prior adoptions

Research across a range of contexts shows that observing prior adoptions by others raises the likelihood that a prospective user will adopt a technology (Terlaak & Gong, 2008). Studies of "information cascades" formalize this social learning mechanism and use it to explain the bandwagon-like adoptions of fashionable innovations (Abrahamson & Fairchild, 1999; Bikhchandani, Hirshleifer, & Welch, 1992) such as Total Quality Management practices (Strang & Macy, 2001). Because actors observe other actors' choices when they interact, the network of social relations between actors affects the frequency of prior adoptions that they perceive (Abrahamson & Rosenkopf, 1997; Davis & Greve, 1997; Greve, 2009).

A hackathon exposes prospective 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 attendees.

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.

H1: The association between attending a temporary gathering and adopting a platform is stronger the higher the proportion of other attendees who have already adopted the platform prior to the temporary gathering.

Social learning from successful outcomes

We expect prospective adopters to be influenced more strongly when they observe a complementor who uses the platform receiving recognition for a successful outcome. Prior work on social learning finds that not all objects of imitation are equally influential (Beckman & Lee, 2017). Awards and endorsements make some actors more prominent (Reschke, Azoulay, & Stuart, 2018; Wade et al., 2006). An observer more likely imitates a prominent actor because the observer pays more attention, and the observer may infer that the actor's prior adoption choices led to the actor's prominence (Denrell, 2003; Rao, Greve, & Davis, 2001).

In the context of platform adoption, prior work finds that the prominence afforded by a "superstar" developer joining a platform can attract additional users to the platform (Binken & Stremersch, 2009). Platform owners themselves can strategically confer prominence on select complements to draw attention to their ecosystem (Rietveld, Schilling, & Bellavitis, 2019). In innovation communities, contributors inspect the projects submitted by other contributors to try to understand the attributes that make a project successful (Riedl & Seidel, 2018). By focusing social learning on certain projects, an award given to one developer is likely to influence the platform choice of other prospective adopters.

We expect that attendees imitate the platform choices of award winners at temporary gatherings. Gatherings ranging from software development hackathons to academic conferences award prizes to select attendees. The award ceremony often takes place in person in front of the other attendees, literally orienting the attendees towards the award-winning project. The technologies used to build the winning project may be subject to heightened attention from other attendees, making them more likely to adopt the technology in their own projects going forward.

H2: The association between attending a temporary gathering and adopting a platform is stronger when an award-winning project at the temporary gathering uses the platform as a component.

Temporary gatherings as forums for knowledge exchange

Prior adopters can also influence prospective adopters through knowledge exchange, i.e., the active sharing of experience on how to adopt and use a technology. Unlike social learning—which entails observation but does not require active effort from the prior adopter being observed—knowledge exchange requires the active participation of both the prospective adopter and the prior adopter. Technology diffusion research recognizes the role of prior adopters in training and coaching prospective adopters in a new technology (e.g., Attewell, 1992). Knowledge exchange could be an important mechanism through which platforms diffuse at temporary gatherings.

Temporary gatherings of various types foster a norm of open knowledge sharing. For example, conferences in academic communities are governed by a scientific norm of common ownership over a shared body of knowledge (Merton, 1973). Software development hackathons often have some affiliation or overlapping membership with open innovation and open-source software communities, which nurture an ethos of openness, valuing the free exchange of ideas and encouraging knowledge to flow between members (Lee & Cole, 2003; Lifshitz-Assaf, 2018; Mahony & Ferraro, 2007; Seidel & Stewart, 2011). Hackathons often adopt norms that encourage the open sharing of expertise (Lifshitz-Assaf, Lebovitz, & Zalmanson, 2020). Many software projects created at hackathons are, themselves, made available as open source. In addition, spatial and temporal proximity at temporary gatherings creates many opportunities for face-to-face communication, which broadens the scope of what knowledge can be shared (Chauvin, Choudhury, & Fang, 2020; Koo & Eesley, 2020; Lee, 2019).

To assess when knowledge exchange might lead to platform diffusion, we consider which prospective adopters might be best positioned to proactively learn from other attendees of a temporary gathering. Building on prior work exploring heterogeneity among complementors (Boudreau & Jeppesen, 2015; Rietveld & Eggers, 2018), we look to the heterogeneity of technical expertise amongst the attendees who are prospective adopters. Technical expertise affects how prospective adopters process information about a new technology, impacting their adoption decision (Patel et al., 2015; Rogers, 1961). For example, a recent study by Greenwood et al. (2019) finds physicians with specialized expertise more rapidly adopt revised guidelines reducing the use of invasive coronary surgery.

There are contrasting lines of reasoning over whether technical expertise makes a prospective adopter more or less likely to be influenced by knowledge exchange. On one hand, technical expertise underlies absorptive capacity (Greenwood et al., 2019; Ter Wal, Criscuolo, & Salter, 2011), meaning attendees with superior technical expertise may be better-positioned to learn from other attendees. This could make them more susceptible to the influence of prior technology adopters at temporary gatherings. On the other hand, technical expertise implies existing familiarity with previously adopted technologies, making the prospective adopter less motivated to learn about a new technology, especially a substitute technology. This could make them less likely to be influenced by prior technology adopters at temporary gatherings. Given these contrasting lines of deductive reasoning, we do not formally hypothesize about how knowledge exchange and technical expertise impact platform diffusion at temporary gatherings. Instead, we empirically investigate this mechanism qualitatively and quantitatively in an exploratory manner.[8]

Temporary gatherings as forums for social coordination

In-person gatherings not only allow attendees to learn from the past adoption choices of others: they also allow attendees to converge on a forward-looking consensus about which technological platform is likely to become dominant in the future. We refer to this mechanism as *social coordination*. In this section, we build on prior research on coordination and social networks to put forward a theory of social coordination through temporary gatherings.

While the social learning mechanisms leading to Hypotheses 1 and 2 apply to a broad range of technologies, social coordination distinctively applies to technologies with network effects, such as platforms or standards (e.g., 5G telecommunications). For technologies with network effects, the value of adoption rises with the number of other users, so prospective users gain if they can coordinate on one platform or standard (Varian & Shapiro, 1998). Platforms can facilitate coordination with governance tactics such as pre-emptive announcements (Farrell & Saloner, 1988) or restricted choice (Casadesus-Masanell & Halaburda, 2014). Social mechanisms for achieving technology coordination have received much less attention (Afuah, 2013).

Temporary gatherings facilitate social coordination by bringing people together while also exposing them to the common influence of a sponsor. Couched in the language of social theory, a temporary gathering serves as a social focus, a term coined by Feld 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). Feld's work depicts social foci as generators of social structure, i.e., spaces in which social ties are established (Feld

[8] Future researchers may be able to make fine-grained distinctions between platform complements and substitutes in order to study platform co-adoption in more detail (Eisenmann, Parker, & Van Alstyne, 2011).

& Grofman, 2011). We expand on this by noting that social foci facilitate coordination in ways that inherently complement platform business models. Platform firms strategically create or sponsor social foci to facilitate coordination. To illustrate, users of e-commerce platforms like 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 known focal points at busy locations like airports where drivers and riders can meet. Social focal points enhance the value of a platform by enabling interactions between platform users and complementors.

Temporary gatherings facilitate a dense network of interactions that go beyond one-to-one “dyadic” conversations: they include group activities as well as centralized audience attention on a speaker (e.g., a keynote speech). In turn, dense networks of interaction permit consensus formation (Farrell & Saloner, 1988; Soh, 2010). As we outline below, our interviews provide evidence that some hackathon attendees are aware of the consensus forming around certain platforms at hackathons. To empirically test the social coordination mechanism, we develop two hypotheses based on this logic of consensus-formation at temporary gatherings.

Social coordination and perceptions of platform dominance

We suggest that the dense network of interactions at a temporary gathering can shift attendee perception over whether a given platform will achieve critical mass, i.e., whether enough developers will adopt a platform to make it dominant in its category (Katz & Shapiro, 1994). Social coordination implies that attendees at a temporary gathering can recognize when a bandwagon is emerging around a new technology and then choose to join it. Separate from social learning about intrinsic platform quality, exposure to prior adopters 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 (Anderson, Parker, & Tan, 2014; Tucker & Zhang, 2010; Zhu & Iansiti, 2012). In the case of platforms for collaborative software development, we assume developers perceive positive direct network effects: 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. As Rogers (2003: 353) puts it, “Individuals adopt an innovation, in part, on the basis of their expectations of others’ future adoption…so *watching while being watched plays an especially important role in the diffusion of interactive innovations*” (emphasis ours).

Perceptions of platform dominance are likely to rise faster-than-linearly with the perceived rate of platform adoption (Arthur, 1989; Eisenmann, 2006). Put concretely, we expect that a shift from a perceived adoption rate from 50% to 75% has a greater impact on perception of platform dominance than a shift from 0% to 25%. This non-linearity follows from the observation that industries with network effects often result in a winner-take-all outcome (Parker & Van Alstyne, 2005), in which a dominant platform is adopted universally while other platforms fall by the wayside.

We do not have any *a priori* reason to propose that one specific adoption threshold matters. However, our theory of social coordination at temporary gatherings allows for a refinement of the earlier hypothesis (H1) linking platform adoption to the rate of prior adoptions by other attendees at the temporary gathering:

> *H3: The association between an attendee adopting the platform and the proportion of other attendees who have already adopted the platform prior to a temporary gathering is non-linear, with a stronger association exhibited at higher proportions of prior adopters.*

Social coordination and sponsor amplification

Coordination may also be achieved when a firm's presence as a sponsor amplifies the social learning processes at a temporary gathering. As depicted in the last row of Figure 1, an attendee interacts simultaneously with other attendees and the sponsor. This set of interactions creates a powerful opportunity for social coordination, accelerating the process through which opinions of a sponsor converge amongst attendees at a temporary gathering.

Balance theory—an established theory in network sociology (Rawlings & Friedkin, 2017)—proposes that people intrinsically prefer for their opinions to be consistent with those of people around them (Heider, 1946). As a result, 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: it sets up a triadic relationship between the two adopters (persons P and O) and the platform (entity X).[9] Balance theory predicts that the resulting triad likely consists of either three positive sentiment ties or exactly two negative sentiment 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 towards X when she sees O positively interacting with X, and *vice versa*.

Because of the psychological pressure towards balanced sentiment, the platform's presence as a sponsor at a temporary gathering likely amplifies the social influence between prior adopters and prospective adopters. The temporary gathering focuses attention on the sponsor, triggering conversations between attendees about their opinion of the sponsor that cause sentiment to converge. This mechanism of opinion formation occurs across contexts. For example, at a political debate, negative sentiment towards a speaker on stage may diffuse

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

throughout the audience (Clayman, 1993). At an academic conference, positive or negative sentiment towards a presented paper may spread throughout an audience. If prior adopters arrive at an event with positive sentiment towards a platform, they likely spread this sentiment to other attendees (H1), and the presence of the platform owner as an event sponsor strengthens the effect of this sentiment.

> *H4: The association between an attendee adopting the platform and the proportion of other attendees who have already adopted the platform prior to a temporary gathering is stronger when the platform owner is a sponsor of the temporary gathering.*

We investigate our theoretical framework using both quantitative and qualitative methods. The following section tests our hypotheses in a new dataset; the subsequent section reports findings from a program of interviews that enrich the interpretation of our quantitative study.

QUANTITATIVE EMPIRICAL METHODS

Our quantitative empirical analysis studies software developers who attended hackathons and measures their adoption of platforms over time. We track the monthly activity of 1,302 software developers from January 2012 to November 2017. Our main dataset consists of a developer-platform-month panel with 786,240 observations. The following subsections outline how we assemble the dataset by selecting a sample of hackathons, identifying developers attending these hackathons, and collecting the corpus of publicly accessible software projects for these developers.

Hackathon sample

We collect data on hackathons—and their organizers, sponsors, and attendees—from the website Devpost.com. 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 firms sponsoring a hackathon. For each event, we record the date, attendees, location, sponsors, and the prize(s) awarded.

We select hackathons for inclusion in our sample based on criteria that align with our theoretical framework and empirical strategy. First, as our theory relates to in-person gatherings, we select hackathons with a physical venue. 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 between several hours and several days.[10] Second, we consider hackathons taking place between January 2014 and May 2017, which corresponds to when hackathons became more prevalent.[11] Third, we select hackathons sponsored by the 29 most-frequent platform sponsors of hackathons events.[12] Fourth, 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 limit developer self-selection into the hackathon sample based on their attraction to a given platform sponsor, which would otherwise bias our results.[13] Finally, we exclude hackathons with fewer than ten identifiable participants because social interaction within a gathering is central to our theoretical framework.[14] The remaining set of 167 hackathons serve as the quasi-experimental treatment events in our study. From Devpost, we identify the developers attending at least one of these hackathons by whether they submitted a project to the hackathon for judging via Devpost.[15] Online Appendix B further details this sample construction.

[10] Hackathons in our sample have durations ranging from eight to 60 hours, with a median of 24 hours.
[11] Online Appendix B depicts the growth over time of hackathons by type.
[12] 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.
[13] We confirm that sponsor information was generally not available prior to the event. From a random 10% sample of our filtered hackathons, we use the Internet Archive to compare the hackathon details before and after each event on Devpost.
[14] Our main findings are robust to an analysis that limits the sample to smaller hackathons with 10 to 15 attendees.
[15] 70% of our developers attended just one in-sample hackathon; 10% attended two in-sample hackathons; and 10% attended three to five hackathons in the sample. 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.

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, an online code repository. Software developers widely use GitHub to publicly share what they are working on and to 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.[16] Many developers, including those in hackathons, use GitHub to collaborate, both within their teams and with the open-source community (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.[17] We omit projects that are copied by the developer from elsewhere, since most of the source code in such instances may not have been written by the developer and they do not represent actual development effort (Wu, Wang, & Evans, 2019). Because our empirical strategy relies on a longitudinal research design and we would like to identify developers who have some baseline of observable development activity, 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 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

[16] 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.

[17] 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. 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; Online Appendix C provides further details on this API identification process. Based on the dates of creation and modifications to a project, 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 variable

The dichotomous variable *Platform Adoption* takes a value of one in the first developer-platform-month in which a developer uses the focal platform, i.e., writes code that calls on a platform's API, and zero if the developer has not yet used the platform (or ever). The developer decision to adopt a platform is a non-repeatable event (Allison, 2014). Thus, after a developer has adopted a platform, subsequent observations of that pair are no longer included in the risk set (Allison & Christakis, 2006).

Independent variables

Hackathon Attendance

Our baseline independent variable indicates 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. This measure only considers the first in-sample hackathon attended by a developer and not any subsequent hackathons.[18]

Local Adoption Rate

To test for whether hackathons act as a forum for social learning 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 platform divided by the total number of peer developers at the hackathon (independent of whether the platform sponsors the hackathon). In months prior to the hackathon, this variable is coded as zero.

Winner Adoption

We test whether successful projects that use the platform at a hackathon event have an influence on prospective adopters. To do so, we identified each project that won an award at the hackathon attended by the focal developer. We then identified the set of platforms used in those projects. For each developer–platform pair, the dichotomous variable *Winner Adoption* takes the value of one in the month of the hackathon and the months that follow for all the platforms used by the award-winning projects, and zero otherwise.

Local Adoption Rate thresholds

We create three indicator variables for different adoption thresholds to test for non-linearity in perceptions of platform dominance across different levels of platform adoption. *Local Adoption > 25%* is set to one if a developer attends a hackathon where the proportion of developers at the hackathon who already adopted a focal platform event exceeds 25%. Similarly, we construct *Local Adoption > 50%* and *Local Adoption > 75%* when the proportion of developers at the event already who already adopted the focal platform exceeds 50% and 75%, respectively.

[18] We perform sub-sample analyses between developers who are only exposed to a single hackathon event and developers who attend multiple hackathon events over the observation window. We find results consistent with our main results for both samples.

Control variables

To address time-variant heterogeneity in developer experience or skill, we include two control variables: for each developer in each month, *Project Experience* and *Platform Stock* measure the stock count of software projects or platforms, respectively, lagged by one month. To control for heterogeneity in the economic inducement offered by a platform sponsor, *Expected Subsidy* is the expected amount (in thousands of US dollars) that a developer could receive from hackathon prizes specifically requiring the use of a sponsor's platform.[19] We expect that larger value of prizes from platform sponsors associates with more adoption of the platform by developers in attendance. We log-transform these three control variables.

Tables 1 and 2 present descriptive statistics and correlations, respectively, for these variables at the platform-developer-month level.

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

Empirical model and strategy

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 at the developer-platform-month level of analysis (Allison, 2014).

We design our sample, model, and variable coverage to minimize the potential for endogeneity; for this type of observational study, we need to address self-selection by developers into hackathons sponsored by platforms they seek to adopt, a form of omitted variable bias. Our data tracks each developer longitudinally before and after attending a platform-sponsored

[19] These platform-specific prizes are distinct from the generic main prize, which does not require the use of any particular platform. We collect information from Devpost about these platform-specific 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 assume each developer perceives their chance of winning a prize to be the same as that of the other developers in attendance.

hackathon, where the hackathons occur on different dates. This design compares outcomes for developer–platform combinations that have previously attended a hackathon sponsored by the platform against developer–platform combinations that have not attended a hackathon sponsored by the platform.

In our main specification, we use conditional hazard models because the developer decision to adopt a platform is a non-repeatable event. Our observations consist of the risk set of instances in which a particular developer has not yet used a given platform. When a developer adopts a platform in a given month, the dependent variable is coded one in that month and the developer–platform dyad then drops out of the risk set in the following months (Allison & Christakis, 2006).

We employ a linear probability model to facilitate interpretation of interaction effects (Greene, 2012). Robust standard errors are clustered by developer. We estimate a number of specifications, building up from a parsimonious model to more specified models, on both the full sample and the various sub-samples. Equation 1 shows the most fully saturated model for developer i, platform j, and month t.

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

All models include the three control variables and platform-month fixed $\gamma_j \times \delta_t$. We seek to address unobserved time-invariant and time-variant heterogeneity in both developers and platforms through fixed effects and control variables. Platform-month fixed effects consist of a fully interacted set of platform fixed effects and time fixed effects and control for substantially more unobserved heterogeneity than separate platform fixed effects and time fixed effects. By addressing general trends in platform use over time (Bothner, 2003), platform-month fixed effects control for changes in the indirect network size and stand-alone value of each platform.[20]

QUANTITATIVE RESULTS

Table 3 presents the regression results that test Hypotheses 1 to 4. Model fit in Table 3—reflected by *Adjusted Within-R^2*—increases from 0.023 in our baseline model to 0.035 as we move from the left to the right of the table, suggesting that the explanatory power increases as we account for more mechanisms.[21]

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

The baseline model (Model 3.0) looks at the association between attending a platform-sponsored hackathon and a developer's hazard of adopting the sponsor's platform. We observe a positive coefficient on *Hackathon Attendance* ($p \sim 0.000$). The coefficient implies a large and meaningful effect size: upon attending a hackathon sponsored by a given platform, the monthly hazard that the developer adopts the platform rises by 1.7 percentage points. Annualized, this is an increase in adoption hazard of 20.4 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 1.8%.

Social learning

Model 3.1 adds the variable *Local Adoption Rate* to test Hypothesis 1. Consistent with the hackathon acting as a forum for social learning (Hypothesis 1), every ten percentage point

[20] Our study of *Platform Adoption* is robust to the inclusion of developer fixed effects to control for time-invariant unobservable heterogeneity across developers. In our main models predicting *Platform Adoption* we omit 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, specifications that include developer-level fixed effects generate results similar to the main reported results. 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. In practice, developer fixed effects capture more granular unobserved heterogeneity than hackathon fixed effects; given that our measures only consider the first in-sample hackathon attended by a developer—implying a strict many-to-one correspondence between developers and hackathons—hackathon fixed effects would be collinear with developer fixed effects.

[21] *Adjusted Within-R^2* accounts for the number of independent variables in the model and excludes the variance explained by the fixed effects.

increment in the *Local Adoption Rate* of a given platform at a hackathon is positively associated with a tenth of a percentage point increase in the monthly hazard of adoption ($p \sim 0.000$).

Model 3.2 adds the variable *Winner Adoption* to test Hypothesis 2. Consistent with Hypothesis 2, when a platform is used in a project that receives an award at the hackathon, the monthly hazard that other developers adopt the platform rises by 0.6 percentage points ($p \sim 0.003$).

Social coordination

Model 3.3 introduces three thresholds of local adoption to test for non-linearity in the local adoption rate (Hypothesis 3). The variable *Local Adoption Rate* is not included in this model because it would be highly collinear with the three indicators. The threshold indicator variables are defined using inequalities rather than mutually exclusive categories: when a higher threshold indicator is set to one, the lower threshold indicators are also, by definition, set to one (e.g., if *Local Adoption > 50%* takes a value of one, that means that *Local Adoption > 25%* also takes a value of one). Each coefficient therefore indicates the additional increase in the monthly hazard rate for values of *Local Adoption Rate* above the threshold and coefficients can be added together to draw a comparison against the baseline, which here is 0% < *Local Adoption Rate* ≤ 25%. There is a 0.1 percentage point increase for those in the lower-middle (25% < *Local Adoption Rate* ≤ 50%) threshold and 0.2 percentage point increase for those in the upper-middle (50% < *Local Adoption Rate* ≤ 75%) threshold, both compared to the baseline. There is an associated 0.9 percentage point increase in the monthly hazard of adopting a platform for those in the highest threshold (75% < *Local Adoption Rate* ≤ 100%) compared to the baseline. Consistent with Hypothesis 3, we find that the association between an attendee adopting the platform and the proportion of other attendees who are prior adopters is non-linear, with a stronger association exhibited at higher proportions of prior adopters, i.e., the relationship is convex.

Model 3.4 tests whether platform diffusion is stronger when there is a sponsor present at the event (Hypothesis 4). The positive coefficient on the interaction term *Local Adoption Rate X Hackathon Attendance* suggests that sponsorship positively moderates platform diffusion, consistent with Hypothesis 4 (p ~ 0.000). For those who attend a hackathon sponsored by a platform, every ten percentage point increment in the *Local Adoption Rate* for the sponsoring platform at a hackathon is associated with a 0.75 percentage point increase in the monthly hazard of adoption.

Additional analyses

Heterogeneity in developer technological expertise

To better understand developer heterogeneity, we categorize developers into three categories based on their technological expertise. For each developer in our sample, we collect data from LinkedIn and GitHub on their skills, education, and programming experience. Using these observable developer characteristics, we apply Latent Class Analysis (LCA) to classify each developer into mutually exclusive categories (Chae, Bruno, & Feinberg, 2019; Rawlings & Friedkin, 2017), which yields three classes of developers, based on developer observables for their technological expertise. *Novice* developers are relatively unskilled, lack formal education, and have little software development experience. *Intermediate* developers are skilled and well-educated, but are inexperienced with software development. *Expert* developers are also skilled and well-educated, but they have strong programming experience. Online Appendix D details the underlying data and construction of these developer categories.

Tables 4a and 4b present the analysis by developer categories. Table 4a runs the same specification as in Model 3.2 of Table 3 by sub-sample. Table 4b examines the differences in coefficients between the different developer groups: expert, intermediate, and novice developers.

We find that prospective adopters with greater technical expertise, i.e., experts, are more likely to respond to social learning compared to the intermediate group. In Model 4b.2, every ten percentage point increment in the *Local Adoption Rate* of a given platform at a hackathon is associated with a 0.07 percentage point increase in the monthly hazard of adoption for experts, compared to intermediate developers (p ~ 0.063). In Model 4b.1, we also see some evidence that experts are more likely to be influenced by award-winning projects compared to novices. Experts experience a one percentage point higher monthly hazard of subsequently adopting a platform compared to novice developers after seeing an award-winning project use the platform at the event (p ~ 0.128). We discuss our interpretation of these results in conjunction with our qualitative findings in the Discussion section of the paper.

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

Social tie formation

In order to directly test for whether hackathons serve as forums for social interaction, we gather data from GitHub. By tracking social activity from GitHub, we are able to examine tie formation between developers who are attending events together. To do this, we collect data on three types of activity from GitHub between 2012 and 2017 to match the observation window for our main analysis. First, we check whether a developer "follows" another developer's profile. Second, we check whether a developer has starred (i.e., bookmarked) any project that is owned by another developer. Finally, we check for whether a developer has directly engaged with another developer's project by joining as a project member, contributing software code to the project, or comments about the project. In total, we observe 220,010 interactions between hackathon users by tracking their GitHub programming activity. Our dataset of social interaction on GitHub directly maps onto the developers whom we track in our main analysis, and each activity between developers has an associated timestamp that allows us to identify when the interaction occurred.

We construct a developer dyad-month panel dataset of 1,200,295 observations from this body of GitHub events. Our sample of developer dyads includes all developers who will eventually attend the same hackathon event in our study. We identify the developer social interaction within a dyad by using the earliest timestamp of activity that involves two developers. The resulting balanced panel data set begins the time series for each developer dyad in January 2012. To be consistent with the main analysis, and to have 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 dyad in November 2017.

Table 5 presents the social tie formation analysis using logit and rare events logit models. We report the odds-ratios estimate: the ratio of the odds of a tie forming between two developers after they attend a hackathon event and the odds of tie formation between the developers prior to the hackathon event. In Model 5.1, we find a 26.3% higher likelihood that a tie forms between two developers after attending an event together ($p \sim 0.000$). In Model 5.2, we collapse our observations from the month-level to the year-level, and we find that there is a 26.0% higher likelihood that a tie will be formed ($p \sim 0.000$). In Model 5.3, we drop monthly observations after December 2016 when GitHub ceased support for developers to star another project, which could lead to a downward bias in our estimates of social interaction. Consistent with this, we find that there is a 30.6% higher likelihood that a tie forms in this sample ($p \sim 0.000$). Given that our data consists of all potential ties that might be formed between developers who attend the same hackathon events, actual tie formation is relatively sparse. To address this, we repeat the same tests using a rare events logit for Models 5.4–5.6, and we find consistent results. Taken together, we find support that temporary gatherings are associated with social interactions between developers.

---------- INSERT TABLE 5 ----------

Robustness checks

We utilize heterogeneity in developer, hackathon, and platform characteristics to further test the robustness of our main findings. Recall that the main alternate explanation for our results is that a developer finds out in advance of a hackathon which platforms will sponsor it, and she attends the hackathon because she planned to adopt the platform anyway, i.e., reverse causality. Although the main model rules out this possibility as best as possible, we run several additional analyses summarized below that provide further confidence for the robustness of the main findings.

Developer pre-hackathon adoption balance test

We check for balance on pre-event observables between developers who attend a hackathon event sponsored by a focal platform (sponsored developers) and developers who attend an event not sponsored by that platform (unsponsored developers). We find that developer pairs in our sponsored and unsponsored groups to be similar on observables prior to hackathon attendance (see Online Appendix E).

Developer geographic proximity instrumental variables analysis

To further address the possibility that developers self-select into hackathons, we estimate two-stage instrumental variable models and find results consistent with the main analysis. The first stage predicts developer attendance at a particular hackathon using the developer's geographic distance from the hackathon as an exogenous instrument.[22] Geographic distance fulfils the characteristics of a desirable instrument because proximity to a hackathon predicts attendance (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

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

(exclusion restriction) (see Online Appendix F).

Platform prominence and media coverage

We consider the possibility that hackathon participants are drawn to the hackathon by the prominence of a platform sponsor, in terms of their sponsorship level and the media coverage of the 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 self-select 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 (see Online Appendix G).

Similarly, we distinguish between platforms with high media coverage and platforms with low media coverage. If developers select into hackathons that have well-known platform sponsors, we would not expect to find an association between hackathon attendance and adoption of a low media coverage platform. We do find that there is a positive effect for both low and high media coverage (see Online Appendix H).

Hackathon duration and type

We examine how heterogeneity in hackathon characteristics might affect our results, specifically their duration and type. With respect to hackathon duration, 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' behavior. Using manually collected data on hackathon duration, we find that both long- and short-duration hackathons have an association with platform adoption, with a larger effect for longer hackathons (see Online Appendix I).

We categorize our set of hackathons into types based on the identity of the organizer: *public* (non-profit or governmental organizer), *trade* (industry group), and *university* hackathons. We analyze each sub-sample of hackathons with the main model and test for differences in coefficient between each type of hackathon. Across all hackathon categories, sponsorship positively moderates the social learning effect (Hypothesis 5) (see Online Appendix J).

QUALITATIVE EVIDENCE

To better understand the empirical setting and to enrich our evidence base, we carried out a program of qualitative interviews with 30 different hackathon stakeholders between May 2019 and August 2019. We interviewed hackathon organizers, sponsors, and developers. Of the organizers, we interviewed those who operated hackathons for industry trade groups, universities, government, non-profit interest groups, and for-profit firms. Our stakeholders were all based in the United States or Canada, and several had experience with hackathons in Asia, Europe, Africa, and Australia. Collectively, our interviewees attended several hundred hackathons (at least), dating from the mid-2000s up to the present day. Online Appendix K documents this qualitative study in detail.

The qualitative evidence supports and augments our quantitative analysis. Our interviewees confirmed that exposure to sponsors at hackathons impacts the platforms they experiment with. A mentor at over 20 hackathons, employed by a large search engine platform that sponsors hackathons, said, “Most developers come in with no idea and no alignment with any other technology. So, they’re picking literally when you’re talking with them.” One developer with experience at six hackathons said, “I learned about Oculus Rift [virtual reality platform] at a hackathon. I didn’t have the device myself, but when I saw it, I knew it was cool. When they loaned me the device to work with, I was like, ‘Hell yeah.’ And I built a Spider-Man game with it.” Another developer reported, “I went to a workshop on using Unity [game engine platform]. It’s pretty tough to get started. They helped us get set up and learn how to use that.”

Although most developers interviewed (11 out of 13) reported that they primarily attended a hackathon to learn new technical skills, some reported being influenced by the hackathon prizes:

> We saw this prize for using Expo IO [mobile application development platform]… So we used Expo to create a mobile app that communicated with our service so you could use your phone to send untranslated images over to our system, and it’ll send back a translated image. We built a translator app that automatically translates text to manga [Japanese cartoon]. And that’s how we won the Nintendo Switch. That was the prize.

Hackathons foster a culture of openness that leads to people helping one another out. A hackathon organizer at a university told us, “The point of our hackathon is not competition. The point is to be in an environment that is fun and positive and collaborative, and it’s kind of like, it’s a win-win for everyone. The point is not to compete directly. The point is to have a good environment where everyone feels good to build something.” An organizer and developer of fifteen hackathons said,

> You’re in this like really big, almost communal space usually. People will quite often end up talking to the people around them. There are lots of workshops where people meet others. There’s also gatherings. We always have a women’s meet up. We also had social events. Like we’d have a game room where people will be playing Super Smash Brothers or ping pong. Even as late as 2:00 A.M., there something happening, some event where you could take a break, go take part and meet people.

Working under time pressure in close spatial proximity forced attendees to interact intensively. An attendee and eventual organizer said of the hackathon environment, “It’s so nice to be working on the technology and then you’re like stuck, and the guy who wrote the [open source] code you’re trying to use is in the building.” An organizer described hackathons thus:

> It really does put some pressure on you to learn on the spot and actually engage in what you are learning. People are depending on you to accomplish something…You’re put in an environment where it’s actually easy to get help from others because you’re comfortable with the fact that you don’t know everything, and there are people around you that do know more about whatever technology.

Just as learning was a key motivation for attending hackathons, our interviewees regularly reported instances in which they had learned from other hackathon participants. Our social learning hypotheses suggest that attendees are influenced by observing others using an API. This received support in our qualitative interviews. A veteran of fifteen hackathons stated, "When I'm walking around, it becomes noticeable what technologies people are using." A mentor said,

> If more people use it, it's a lower risk because it will be useable. If only one other team is using one API, how do you know if it's working or if this team was just lucky to get it running? Kind of like early adopters versus later adopters kind of thing. Now there's only a certain percentage of the population that are early adopters, so they get a high from using technology for the first time. But most people just follow others.

Another developer noted, "As a developer, you really want to minimize your risk with APIs. And a new API is just like new surface area where like things can go wrong. And so the more well-known it is and the more that there are people around you who have tried it and been successful, the more confident you are that it'll be fine when you use it." Some developers also shared their experience teaching others about new platforms, including a developer who eventually organized hackathons too:

> Twilio is something that I've recommended to people at a hackathon. Like they had a great API. It was a good experience, so if it fit the bill, I would say, 'Hey, this was actually really easy to integrate. Try out this API if it fits the bill.' I definitely defended [Microsoft] Azure to people based on my experience, saying 'You might be more familiar with AWS, but it's actually pretty much the same, I think that it is worth giving it a try.'

Consistent with Hypothesis 2, interviewees report paying close attention to projects that win prizes. One developer stated, "If it's a grand prize winner, that would be something like, wow, I want to do that, too. So right after I get home, I look into the technology they used to build their app."

In addition to learning from observation, an important form of learning was the explicit coaching that participants provided to each other. Hackathon attendees frequently reported learning how to use specific APIs from their teammates. A novice developer remarked, “I was more familiar with like the whole front-end design thing for [Google] Chrome [web browser platform] and my other teammates were more sort of like on the back-end SQL [database platform]. So we just exchanged skills there.” Perhaps surprisingly, participants reported extensive knowledge exchanges between members of competing teams. A developer noted, “We were housed in this room with a couple other teams and we had a thing we needed to debug. One of the guys on the other team overheard, and he just came over and helped a bit. So, that was quite chill. Most people are not very competitive… I’d say most of the time everyone’s supportive.” A hackathon community leader noted, “The density of knowledge is high enough that you could stand up and just yell out, ‘Hey, who knows Arduino [hardware prototyping platform]?’ And there is a high likelihood someone can come help you with that.” Another developer noted, “We would talk about our app with other teams. We got advice from other teams and talk about what we were doing and stuff. They gave us advice and we gave other teams advice.” This peer-to-peer learning was particularly beneficial to more experienced software developers. One hackathon organizer told us,

> The best developers, the ones gunning for the prize, know that they can get whatever working by the end of the hackathon, so they would take more chances on a new technology which is hard because there is also less documentation they can find on Stack Overflow [developer Q&A forum]. But they also knew what questions they needed to ask when they hit roadblocks.

Our theoretical development argues that hackathons are forums for social coordination. While our interviewees did not talk explicitly in terms of network effects and platform dominance, several of our interviewees did refer to the impact of what other developers at the hackathon were using. A developer said, “It was better to use Ruby on Rails than some obscure platform because of all the resources available around us. We need to be able to put it into practice right away.” A developer noted,

> It’s much more noticeable at a hackathon when people have confidence in a certain technology ’cause you will hear people talk about it, and you will see like lots of people talking to a certain company about their technology. And that makes it salient in way that, like no individual is really usually tracking like the popularity of random technologies on the Internet. You’re not doing that. You’re only looking up a question when you have them about the thing that you care about in that moment.

A well-known platform evangelist observed,

> There is a herd mentality because you figure that the other people are the experts. Everyone, especially in tech there is a large degree—some people describe it as a cluster syndrome. Everywhere I look at the hackathon, everyone’s talking about this technology. And look at all of these people that know it and they’re very happy about it. So yeah, like if it works for them, maybe this is what I should be doing. A lot of people haven’t really formed the ability to just look past the marketing or the inertia of a certain thing. This is why when new technologies come out, you just see so much inertia because everyone has a fear of missing out. This is what everyone is doing so, it’s what I need to be doing.

This helps substantiate our arguments for Hypotheses 3 and 4 that hackathons can catalyze a virtuous cycle of platform adoption, especially when the platform is present as a sponsor to tip the crowd’s sentiment in their favor.

DISCUSSION

Our quantitative and qualitative evidence indicates that temporary gatherings such as hackathons act as an important forum for the diffusion of platform technology, and they allow platform owners to attract complementors to their ecosystem. We document four distinct channels through which platforms diffuse at temporary gatherings (see Figure 1). First, when the platform owner is present as a sponsor of the temporary gathering, this generates a baseline level of awareness and provides an economic inducement that leads some attendees to adopt the platform. Second, temporary gatherings are forums for social learning: attendees make inferences about platform quality by observing how frequently other developers use a particular platform and which platforms are used by the most successful (i.e., award-winning) peers.

Third, attendees can actively exchange knowledge about how to adopt and use platforms. We find that attendees with greater technical expertise are more likely to adopt a technology based on what other attendees at a temporary gathering use, suggesting that an existing knowledge base positions a prospective adopter to learn more readily from knowledge exchange with their peers. Our qualitative findings provide further detail on this mechanism. Interviewees reported that experienced developers were willing to take chances on a new technology because they had the confidence that their general knowledge gave them a clearer sense of problems that might arise and enabled them to ask the right the questions of their peers when they do, reflecting an enhanced ability to leverage knowledge exchange for learning.

Fourth, temporary gatherings are forums for social coordination: they allow attendees to converge on a forward-looking consensus about which technological platform is likely to become dominant in the future. We find that platform adoption hazard rises non-linearly with the proportion of attendees who are prior platform adopters, with a stronger effect when the proportion of other adopters is high; this is consistent with attendees using hackathons to identify emerging technology bandwagons. We also find the presence of a platform owner as a hackathon sponsor strengthens the association between adoption hazard and the proportion of attendees who are prior adopters, consistent with the convergence of sentiment towards the platform predicted by balance theory. Table 6 summarizes the evidence we find for the three social channels of platform diffusion: social learning, knowledge exchange, and social coordination.

---------- INSERT TABLE 6 ----------

Our work helps address a gap in the literature highlighted in a recent review: “Despite the importance of complementors to platform success… very little attention has been paid to the antecedents of complementor support” (McIntyre & Srinivasan, 2017: 155). We build on the foundations of prior research on the formation of platform–complementor relationships, which predominantly studies formal mechanisms (e.g., Hagiu, 2006) by identifying informal antecedents of complementor support, i.e., social learning and social coordination. We thus contribute to an emerging stream of work that takes a socially embedded view of complementor behavior (Boudreau & Jeppesen, 2015; Eckhardt, 2016; Mollick, 2016; Nagaraj & Piezunka, 2018). We advance a novel methodology for studying platform–complementor relationships by demonstrating one way to use open-source code to study developers longitudinally (Baldwin, Maccormack, & Rusnak, 2014; Polidoro & Yang, 2019). Our study has ramifications for two major areas of strategy research: how coordination enables ecosystem emergence and how social structure affects technology diffusion.

Temporary gatherings, social coordination, and ecosystem emergence

An important question for both theory and practice of platform strategy is how a platform can attract complementors. Prior research studies mechanisms such as price subsidies and architectural and governance choices, which can motivate a complementor to join or leave a platform (Armstrong, 2006; Chu & Wu, 2018; Gu & Zhu, 2020). In this paper, we show that, in a setting with poor visibility, temporary gatherings such as hackathons shape the platform adoption decisions of complementors: an attendee is more likely to adopt a platform which sponsors the gathering, which was previously adopted by more other attendees, and which is used by another attendee in a prize-winning project at the gathering. These findings complement existing work that highlights the value of temporary gatherings for educating complementors about platform capabilities (Özalp et al., 2018; West & Wood, 2013) and reducing uncertainty around an ecosystem's trajectory (Dattée et al., 2018; Grodal, 2018).

Our study finds that complementor choices are influenced not just by awareness, subsidies, and social learning, but also by the mechanism of social coordination. Our view of social coordination builds on past research that likens the orchestration of a platform ecosystem to a coordination game: despite the heterogeneous interests of individual actors, all actors prefer to reach a coordinated equilibrium over remaining uncoordinated (Hagiu & Spulber, 2013; Halaburda & Yehezkel, 2016). Bringing people together in a face-to-face social context and exposing them to a centralized influence (e.g., a prize-giving presentation) can help establish the platform as the most obvious tool—the "go-to" or default software component—to solve a given problem (e.g., making Google Maps the "go-to" mapping API). Influencing complementors in this way creates value by solving the technological coordination game using a "focal point," a cognitive solution to achieving coordination outlined by Thomas Schelling (1980). We argue that this concept deserves revisiting in the context of orchestrating platform-based ecosystems (see also Halaburda & Yehezkel, 2019). Our study therefore contributes to the broader literature on how innovation ecosystems achieve coordination (Clough, 2017; Farrell & Simcoe, 2012; Leiponen, 2008; Miller & Toh, 2020).

As a distinctive mode of ecosystem coordination, social coordination at a temporary gathering may apply broadly as a platform growth strategy (Wu, Clough, & Kaletsky, 2019). Anecdotal evidence suggests temporary gatherings may be used to bring users, not just complementors, onboard a platform. 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.

The mechanism of social coordination helps explain why temporary gatherings are powerful forums for behavioral change: they harness 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 and the platform (Rawlings & Friedkin, 2017).[23]

Social structure and platform diffusion

Technology diffusion has been of interest to practitioners and scholars of multiple disciplines for close to a century (Rogers, 2003). Early work in this field of research recognizes that interpersonal social influence, mediated by social networks, is central to diffusion (Burt, 1987; Coleman, Katz, & Menzel, 1957; but cf. Van Den Bulte & Lilien, 2001). Our study contributes to three contemporary debates in this literature: whether strong ties or weak ties are more important conduits for technology diffusion (Aral, 2016; Aral & Van Alstyne, 2011); whether contagions require single or multiple sources of influence to spread (Centola & Macy, 2007; Song & Gargiulo, 2019); and how firms can strategically leverage social networks to seed the diffusion of a product (e.g., Banerjee et al., 2013; Galeotti & Goyal, 2009).

First, where prior research considers the relative importance of the prospective adopter's strong ties and weak ties for their technology choices (e.g., Suarez, 2005), our study draws attention to their transient interactions with strangers. Simmel (1950), in his foundational writings on social structure, described how relationships with strangers are impartial as strangers are unencumbered by prior commitments. This could make strangers unexpectedly effective vectors for technology diffusion as long as initial distrust is overcome (Rogers, 1999). However, because interactions between strangers are difficult to measure, they are at risk of being overlooked by researchers (Brashears & Quintane, 2018). Complementing the growing body of research on social influence in anonymous online communities (e.g., Centola, 2010), our study advances the idea that with the right set of contextual cues (e.g., an open innovation culture),

[23] Interestingly, the theoretical logic of social coordination also highlights how a single *unsuccessful* event will put off a large proportion of potential platform adopters all at once. The 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).

interactions between strangers, as distinct from strong tie or weak tie acquaintances, can be a vehicle for technology diffusion.

Second, our findings support the proposition that platform diffusion can be characterized as a complex contagion, defined as a behavior, practice, or technology that spreads through "independent affirmation or reinforcement from multiple sources" (Centola & Macy, 2007: 703). We find that the relationship between the likelihood that a hackathon attendee will adopt a platform and the proportion of attendees who have previously adopted it is positive and strongly convex, i.e., multiple sources of influence are especially impactful. We contribute to literature on complex contagions by identifying a new mechanism through which these contagions spread. Prior literature suggests that complex contagions spread between social clusters when there are multiple social ties, known as "wide bridges," connecting those clusters (Centola, 2015; Cohen, Hsu, & Dahlin, 2016). Complex contagions can therefore spread via spatial diffusion because adjacent geographic areas are linked by wide bridges (Davis & Greve, 1997; Liu, King, & Bearman, 2010; Toole, Cha, & González, 2012). Our study shows that complex contagions can also spread at temporary gatherings: focused interactions with strangers in a delimited timeframe can create the multiple sources of influence that encourage a prospective adopter to try out a new technology. Technologies can "jump" between network clusters when prior adopters are brought into contact with prospective adopters at temporary gatherings, especially if the technology owner is present as a sponsor at the event. Future work could examine the role of other types of temporary events—such as academic conferences, tournaments, and world's fairs—in the historical diffusion of technologies.

Third, we provide a novel perspective on the role of managerial agency in promoting the diffusion of an innovation. Recent research focuses on whether managers can seed diffusion by identifying influential nodes within a network and subsidizing their adoptions in a targeted way

(Aral & Walker, 2012). While potentially powerful, this approach takes the underlying network itself as a given. Our study finds that hackathons act as social contexts where attendees form ties with one another, echoing recent findings on the formation of collaboration ties at academic conferences (Chai & Freeman, 2019). By sponsoring a temporary gathering, a manager agentically intervenes in the underlying network, helping to create a new social focus around which people can form ties (Kneeland & Kleinbaum, 2020; Lomi et al., 2014). A manager can use a temporary event to bring prospective adopters into contact both with the firm's platform and with prior platform adopters, who then influence the prospective adopter's behavior.

Limitations and future directions

We recognize three limitations of the present study that could lead to future research. First, we are unable to conclusively test the direction of causality in the association between hackathon attendance and platform adoption. We have constructed our sample in a way that minimizes the possibility of developers self-selecting into hackathons because they know the sponsor and they already plan to try out the sponsor's platform. Online Appendix F reports an instrumental variable analysis that uses a developer's geographic location as an exogenous instrument; the results of this analysis are consistent with our main findings. Nevertheless, to provide stronger causal identification, future researchers might consider studying platform diffusion at software hackathons by running a field experiment that randomizes attendees' exposure to hackathon sponsors or assigns attendees to teams in order to randomize peer influence effects (Catalini & Tucker, 2017; Ghosh & Wu, 2020; Greenberg, 2018).

Second, we are not able to conclusively separate the distinct causal channels through which social dynamics at hackathons might affect platform adoption. Most materially, we are not able to quantify the relative importance of frequency-based imitation and interpersonal knowledge exchange. Theoretically, the two mechanisms are slightly different; however, the variable we use in our test of frequency-based imitation could also be a proxy for interpersonal knowledge exchange. Our qualitative investigation gives us confidence that both mechanisms occur in parallel. Even so, future research might be able to disentangle the relative importance of the two mechanisms. For example, researchers could use non-intrusive methods such as smart badges to measure the extent of person-to-person communication at a hackathon (Matusik et al., 2019). Rich conversational data could capture both knowledge exchange and sentiments expressed towards certain platforms, permitting rich analysis of the mechanisms underlying platform diffusion (Choudhury et al., 2019).

Third, in our main analyses, we are unable to measure the granular structure of interactions between attendees at the temporary gathering. When we use the proportion of attendees who are prior adopters (*Local Adoption Rate*) as an explanatory variable, we, in effect, assume each pair of attendees has an equal probability of influencing one another. To alleviate this concern, our supplementary analyses measure dyad-level interactions with digital trace data. Future research could employ temporary gatherings planned by the researcher (e.g., Camuffo et al., 2020) to examine granular interaction structures in order to study phenomena such as decision aggregation and knowledge recombination in teams (Aggarwal, Hsu, & Wu, 2020; Piezunka, Aggarwal, & Posen, 2020; Wu, 2016) and the contribution of social learning to exploratory search (e.g., Csaszar & Siggelkow, 2010; Fang, Lee, & Schilling, 2010).

CONCLUSION

With this paper, we put forward a theory of social coordination around a platform technology. We show that platform owners can cultivate an ecosystem of complementors by engaging with prospective adopters at temporary gatherings. In their recent review, McIntyre and Srinivasan (2017) identify three prevailing perspectives on platforms, grounded in 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, which identify temporary gatherings as forums for tie formation and social change (Breiger, 1974; Rao & Dutta, 2012). Given the growing interest amongst scholars and practitioners in how managers mobilize resources to build ecosystems (Clough et al., 2019; Seidel & Greve, 2017) and how networks and ecosystems interact (Shipilov & Gawer, 2020), we are optimistic that sociological and organizational literatures have much to offer to our understanding of platform ecosystem management. We believe that our paper opens up a broad line of enquiry into the role of social coordination in ecosystem emergence.

ACKNOWLEDGEMENTS

The following individuals provided helpful feedback on earlier versions of the manuscript: Kevin Boudreau, Sourobh Ghosh, Shane Greenstein, Grace Gu, Marco Iansiti, Daniel Keum, Do Yoon Kim, Wesley Koo, Sunkee Lee, Henning Piezunka, Marc-David Seidel, Balagopal Vissa, and Feng Zhu. Seminar participants at the University of British Columbia, University of Connecticut, Harvard University, Simon Fraser University, Academy of Management Annual Meeting, Strategic Management Society Annual Conference, Strategy Science Conference, and Wharton Technology and Innovation Conference also provided important suggestions. Jaeho Kim, Raghav Pemmireddy, Eric Wu, and Kevin Wu provided excellent research assistance. Jeff Strabone provided copyediting support. This research was funded in part by the HBS Division of Research and Faculty Development and the Social Sciences and Humanities Research Council of Canada.

REFERENCES

Abrahamson E, Fairchild G. (1999). Management fashion: Lifecycles, triggers, and collective learning processes. *Administrative Science Quarterly 44*(4), 708–740.

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

Adner R. (2017). Ecosystem as structure: An actionable construct for strategy. *Journal of Management 43*(1), 39–58.

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

Aggarwal VA, Hsu DH, Wu A. (2020). Organizing knowledge production teams within firms for innovation. *Strategy Science 5*(1), 1–16.

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.

Aral S. (2016). The future of weak ties. *American Journal of Sociology 121*(6), 1931–1939.

Aral S, Van Alstyne M. (2011). The diversity-bandwidth trade-off. *American Journal of Sociology 117*(1), 90–171.

Aral S, Walker D. (2012). Identifying influential and susceptible members of social networks. *Science 337*, 337–341.

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.

Armstrong M. (2006). Competition in two-sided markets. *RAND Journal of Economics 37*(3), 668–691.

Arthur WB. (1989). Competing technologies, increasing returns, and lock-in by historical events. *Economic Journal 99*(394), 116–131.

Assenova VA. (2018). Modeling the diffusion of complex innovations as a process of opinion formation through social networks. *PLOS ONE 13*(5), 1–18.

Attewell P. (1992). Technology diffusion and organizational learning: The case of business computing. *Organization Science 3*(1), 1–19.

Baldwin C, Maccormack A, Rusnak J. (2014). Hidden structure: Using network methods to map system architecture. *Research Policy 43*(8), 1381–1397.

Bandura A. (1986). *Social foundations of thought and action: A social cognitive*. Prentice-Hall: Englewood Cliffs, NJ.

Banerjee A, Chandrasekhar AG, Duflo E, Jackson MO. (2013). The diffusion of microfinance. *Science 341*(6144).

Beaman LA, BenYishay A, Magruder J, Mobarak AM. (2018). Can network theory-based targeting increase technology adoption? *NBER Working Paper 24912*.

Beckman CM, Lee HJ. (2017). Social comparison and learning from others. In *Oxford Handbook of Group and Organizational Learning*, Argote L, Levine JM (eds). Oxford University Press: Oxford.

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.

Binken JLG, Stremersch S. (2009). The effect of superstar software on hardware sales in system markets. *Journal of Marketing 73*(2), 88–104.

Bothner MS. (2003). Competition and social influence: The diffusion of the sixth-generation processor in the global computer industry. *American Journal of Sociology 108*(6), 1175–

1210.
Boudreau KJ. (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.
Brandenburger AM, Nalebuff BJ. (1996). *Co-opetition*. Currency Doubleday: New York.
Brashears ME, Quintane E. (2018). The weakness of tie strength. *Social Networks 55*, 104–115.
Breiger RL. (1974). The duality of persons and groups. *Social Forces 53*(2), 181–190.
Van Den Bulte C, Lilien GL. (2001). Medical innovation revisited: Social contagion versus marketing effort. *American Journal of Sociology 10*(5), 1409–1435.
Burt RS. (1987). Social contagion and innovation: Cohesion versus structural equivalence. *American Journal of Sociology 92*(6), 1287–1335.
Camuffo A, Cordova A, Gambardella A, Spina C. (2020). A scientific approach to entrepreneurial decision making: Evidence from a randomized control trial. *Management Science 66*(2), 564–586.
Cartwright D, Harary F. (1956). Structural balance: A generalization of Heider's theory. *Psychological Review 63*(5).
Casadesus-Masanell R, Halaburda H. (2014). When does a platform create value by limiting choice? *Journal of Economics and Management Strategy 23*(2), 259–293.
Catalini C, Tucker C. (2017). When early adopters don't adopt. *Science 357*(6347), 135–136.
Ceccagnoli M, Forman C, Huang P, Wu DJ. (2012). Co-creation of value in a platform ecosystem: The case of enterprise software. *MIS Quarterly 36*(1), 263–290.
Cennamo C, Özalp 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.
Centola D. (2010). The spread of behavior in an online social network experiment. *Science 329*(5996), 1194–1197.
Centola D. (2015). The social origins of networks and diffusion. *American Journal of Sociology 120*(5), 1295–1338.
Centola D. (2018). *How behavior spreads: The science of complex contagions*. Princeton University Press: Princeton.
Centola D, Macy M. (2007). Complex contagions and the weakness of long ties. *American Journal of Sociology 113*(3), 702–734.
Chae I, Bruno HA, Feinberg FM. (2019). Wearout or weariness? Measuring potential negative consequences of online ad volume and placement on website visits. *Journal of Marketing Research 56*(1), 57–75.
Chai S, Freeman RB. (2019). Temporary colocation and collaborative discovery: Who confers at conferences. *Strategic Management Journal 40*(13), 2138–2164.
Chauvin J, Choudhury P, Fang TP. (2020). The effects of temporal distance on communication patterns in a large multinational: Evidence from daylight savings time. Presented at the annual meeting of the Strategic Management Society.
Choudhury P, Wang D, Carlson NA, Khanna T. (2019). Machine learning approaches to facial and text analysis: Discovering CEO oral communication styles. *Strategic Management Journal 40*(11), 1705–1732.

Chu LY, Wu B. (2018). Designing online platforms for offline services: The bigger, the better? *SSRN* 1–32. Available at: https://dx.doi.org/10.2139/ssrn.3221791.

Church J, Gandal N. (1992). Network effects, software provision, and standardization. *Journal of Industrial Economics 40*(1), 85–103.

Clayman SE. (1993). Booing: The anatomy of a disaffiliative response. *American Sociological Review 58*(1), 110–130.

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.

Clough DR. (2017). Do alliance networks solve the ecosystem timing dilemma? Evidence from the mobile telco industry. Presented at the annual meeting of the Academy of Management. Atlanta, GA.

Clough DR, Fang TP, Vissa B, Wu A. (2019). Turning lead into gold: How do entrepreneurs mobilize resources to exploit opportunities? *Academy of Management Annals 13*(1), 240–271.

Clough DR, Piezunka H. (2020). Tie dissolution in market networks: A theory of vicarious performance feedback. *Administrative Science Quarterly* (Forthcoming), 1–46.

Cohen SK, Hsu ST, Dahlin KB. (2016). With whom do technology sponsors partner during technology battles? Social networking strategies for unproven (and proven) technologies. *Organization Science 27*(4), 846–872.

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

Csaszar FA, Siggelkow N. (2010). How much to copy? Determinants of effective imitation breadth. *Organization Science 21*(3), 661–676.

Cusumano MA, Gawer A. (2002). *Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation*. Harvard Business School Press: Boston.

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.

Davis GF, Greve HR. (1997). Corporate elite networks and governance changes in the 1980s. *American Journal of Sociology 103*(1), 1–37.

Denrell J. (2003). Vicarious learning, undersampling of failure, and the myths of management. *Organization Science 14*(3), 227–243.

Denrell J, Fang C, Winter SG. (2003). The economics of strategic opportunity. *Strategic Management Journal 24*(10), 977–990.

DiMaggio PJ, Powell WW. (1983). The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review 48*(2), 147–160.

Dougherty D, Dunne DD. (2011). Organizing ecologies of complex innovation. *Organization Science 22*(5), 1214–1223.

Eckhardt JT. (2016). Welcome contributor or no price competitor? The competitive interaction of free and priced technologies. *Strategic Management Journal 37*(4), 742–762.

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.

Fang C, Lee J, Schilling MA. (2010). Balancing exploration and exploitation through structural design: The isolation of subgroups and organizational learning. *Organization Science 21*(3),

625–642.
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: 1967–2072.
Farrell J, Saloner G. (1988). Coordination through committees and markets. *RAND Journal of Economics 19*(2), 235–252.
Farrell J, Simcoe T. (2012). Choosing the rules for consensus standardization. *RAND Journal of Economics 43*(2), 235–252.
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.
Flores M, Golob M, Maklin D, Tucci C. (2019). Speeding-up innovation with business hackathons: Insights into three case studies. *Conference Proceedings of the Academy for Design Innovation Management 2*(1), 656–677.
Frolund L, Murray F, Riedel M. (2018). Developing successful strategic partnerships with universities. *MIT Sloan Management Review 59*(2), 71–79.
Gaba V, Dokko G. (2016). Learning to let go: Social influence, learning, and the abandonment of corporate venture capital practices. *Strategic Management Journal 37*, 1558–1577.
Galeotti A, Goyal S. (2009). Influencing the influencers: A theory of strategic diffusion. *RAND Journal of Economics 40*(3), 509–532.
Garud R. (2008). Conferences as venues for the configuration of emerging organizational fields: The case of cochlear implants. *Journal of Management Studies 45*(6), 1061–1088.
Gawer A, Henderson RM. (2007). Platform owner entry and innovation in complementary markets: Evidence from Intel. *Journal of Economics and Management Strategy 16*(1), 1–34.
Ghosh S, Wu A. (2020). Iterative coordination and innovation. *Harvard Business School Working Paper No. 20-121*.
Gousios G, Spinellis D. (2014). Conducting quantitative software engineering studies with Alitheia Core. *Empirical Software Engineering 19*(4), 885–925.
Granovetter M. (1973). The strength of weak ties. *American Journal of Sociology 78*(6), 1360–1380.
Granovetter M. (1978). Threshold models of collective behavior. *American Journal of Sociology 83*(6), 1420–1443.
Greenberg J. (2018). A novel experimental test of social network opportunity and structure in entrepreneurial pitch evaluation updating. *SSRN*. Available at: http://dx.doi.org/10.2139/ssrn.3129431.
Greene WH. (2012). *Econometric analysis*, 7th ed. Pearson Prentice Hall: Upper Saddle River, NJ.
Greenwood BN, Agarwal R, Agarwal R, Gopal A. (2019). The role of individual and organizational expertise in the adoption of new practices. *Organization Science 30*(1), 191–213.
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.
Grodal S. (2018). Field expansion and contraction: How communities shape social and symbolic boundaries. *Administrative Science Quarterly 63*(4), 783–818.

Gu Y, Zhu F. (2020). Trust and disintermediation: Evidence from an online freelance marketplace. *Management Science* (Forthcoming), 1–14.
Gulati R, Sytch M, Tatarynowicz A. (2012). The rise and fall of small worlds : Exploring the dynamics of social structure. *Organization Science 23*(2), 449–471.
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 and Management Strategy 28*, 49–59.
Hannah DP, Eisenhardt KM. (2018). How firms navigate cooperation and competition in nascent ecosystems. *Strategic Management Journal 39*(12), 3163–3192.
Haunschild PR, Miner AS. (1997). Modes of interorganizational imitation: The effects of outcome salience and uncertainty. *Administrative Science Quarterly 42*(3), 472–500.
Haveman HA. (1993). Follow the leader: Mimetic isomorphism and entry into new markets. *Administrative Science Quarterly 38*(4), 593–627.
Heider F. (1946). Attitudes and cognitive organization. *Journal of Psychology*. Routledge *21*(1), 107–112.
Hogg MA. (2010). Influence and Leadership. In *Handbook of Social Psychology*, Fiske ST, Gilbert DT, Lindzey G (eds). John Wiley & Sons: Hoboken, NJ: 1166–1207.
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.
Ingram P, Morris MW. (2007). Do people mix at mixers? Structure, homophily, and the "life of the party". *Administrative Science Quarterly 52*, 558–585.
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.
Katz ML, Shapiro C. (1994). Systems competition and network effects. *Journal of Economic Perspectives 8*(2), 93–115.
Kneeland MK, Kleinbaum AM. (2020). On agency and its limits: The asymmetric effects of offsites on network tie formation. *SSRN*. Available at: https://dx.doi.org/10.2139/ssrn.3520640.
Koo WW, Eesley CE. (2020). Rural e-commerce enhanced by return talent: Evidence from Alibaba. *INSEAD Working Paper*.
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.
Lee GK, Cole RE. (2003). From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science 14*(6).
Lee S. (2019). Learning-by-moving: Can reconfiguring spatial proximity between organizational members promote individual-level exploration? *Organization Science 30*(3), 467–488.

Leiponen AE. (2008). Competing through cooperation: The organization of standard setting in wireless telecommunications. *Management Science 54*(11), 1904–1919.
Lemley MA. (2006). Terms of use. *Minnesota Law Review 624*, 459–483.
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.
Lifshitz-Assaf H, Lebovitz S, Zalmanson L. (2020). Minimal and adaptive coordination: How hackathons' projects accelerate innovation without killing it. *Academy of Management Journal* (Forthcoming).
Liu KY, King M, Bearman PS. (2010). Social influence and the autism epidemic. *American Journal of Sociology 115*(5), 1387–1434.
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.
Mahony SO, Ferraro F. (2007). The emergence of governance in an open source community. *Academy of Management Journal 50*(5), 1079–1106.
Matusik JG, Heidl R, Hollenbeck JR, Yu A, Lee HW, Howe M. (2019). Wearable bluetooth sensors for capturing relational variables and temporal variability in relationships: A construct validation study. *Journal of Applied Psychology 104*(3), 357–387.
McIntyre DP, Srinivasan A. (2017). Networks, platforms, and strategy: Emerging views and next steps. *Strategic Management Journal 38*(1), 141–160.
Mead S. (2012), June 12. How Reddit got huge: Tons of fake accounts. *Vice*.
Merton RK. (1973). *The sociology of science: Theoretical and empirical investigations*. University of Chicago Press: Chicago.
Miller CD, Toh PK. (2020). Complementary components and returns from coordination within ecosystems via standard setting. *Strategic Management Journal* 1–36.
Mollick E. (2016). Filthy lucre? Innovative communities, identity, and commercialization. *Organization Science 27*(6), 1472–1487.
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 No. 2018/05/EFE*. Available at: http://dx.doi.org/10.2139/ssrn.3043095.
Özalp H, Cennamo C, Gawer A. (2018). Disruption in platform-based ecosystems. *Journal of Management Studies 55*(7), 1203–1241.
Parker GC, Van Alstyne MW, Jiang X. (2017). Platform ecosystems: How developers invert the firm. *MIS Quarterly 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.
Patel PC, Kohtamaki M, Parida V, Wincent J. (2015). The effect of firm compensation structures on the mobility and entrepreneurship of extreme performers. *Strategic Management Journal 36*, 1739–1749.
Piezunka H, Aggarwal VA, Posen H. (2020). Learning-by-participating: The dual role of structure in aggregating information and shaping learning. *INSEAD Working Paper No. 2019/29/EFE*.
Polidoro F, Yang W. (2019). Venture growth and multi-homing expansion: Evidence from open source platform complementors. *Academy of Management Proceedings 2019*(1), 14262.
Rao H, Dutta S. (2012). Free spaces as organizational weapons of the weak: Religious festivals and regimental mutinies in the 1857 Bengal Native Army. *Administrative Science Quarterly 57*(4), 625–668.
Rao H, Greve HR, Davis GF. (2001). Fool's gold: Social proof in the initiation and abandonment of coverage by Wall Street analysts. *Administrative Science Quarterly 46*(3), 502–526.

Rawlings CM, Friedkin NE. (2017). The structural balance theory of sentiment networks: Elaboration and test. *American Journal of Sociology 123*(2), 510–548.
Reschke BP, Azoulay P, Stuart TE. (2018). Status spillovers: The effect of status-conferring prizes on the allocation of attention. *Administrative Science Quarterly 63*(4), 819–847.
Riedl C, Seidel VP. (2018). Learning from mixed signals in online innovation communities. *Organization Science 29*(6), 1010–1032.
Rietveld J, Eggers JP. (2018). Demand heterogeneity in platform markets: Implications for complementors. *Organization Science 29*(2), 304–322.
Rietveld J, Schilling MA, Bellavitis C. (2019). Platform strategy: Managing ecosystem value through selective promotion of complements. *Organization Science 30*(6), 1232–1251.
Rochet J, Tirole J. (2003). Platform competition in two-sided markets. *Journal of the European Economic Association 1*(4), 990–1029.
Rogers EM. (1961). Characteristics of agricultural innovators and other adopter categories. *Research Bulletin 882*, 1–66.
Rogers EM. (1999). Georg Simmel's concept of the stranger and intercultural communication research. *Communication Theory 9*(1), 58–74.
Rogers EM. (2003). *Diffusion of innovations*. Free Press: New York.
Rogers EM, Kincaid DL. (1981). *Communication networks: Toward a new paradigm for research*. Free Press: New York.
Schelling TC. (1978). *Micromotives and macrobehavior*. W.W. Norton: New York.
Schelling TC. (1980). *The strategy of conflict*. Harvard University Press: Cambridge, MA.
Schilling MA. (2003). Technological leapfrogging: Lessons from the U.S. video game console industry. *California Management Review 45*(3), 5–32.
Seamans R. (2012). Fighting city hall: Entry deterrence and new technology deployment in local cable TV markets. *Management Science 58*(3), 461–475.
Seidel M-DL, Greve HR. (2017). Emergence: How novelty, growth, and formation shape organizations and their ecosystems. In *Emergence*. Emerald Publishing: Bingley, 50: 1–27.
Seidel M-DL, Stewart KJ. (2011). An initial description of the C-form. *Research in the Sociology of Organizations 33*, 37–72.
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.
Shipilov A, Gawer A. (2020). Integrating research on interorganizational networks and ecosystems. *Academy of Management Annals 14*(1), 92–121.
Simmel G, Wolff KH. (1950). *The Sociology of Georg Simmel*. Free Press: Glencoe, IL.
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.
Soh P-H. (2010). Network patterns and competitive advantage before the emergence of a dominant design. *Strategic Management Journal 31*, 438–461.
Song JS, Gargiulo M. (2019). David vs. Goliath in the digital age: The effect of network structure and content on the adoption of cultural products. *INSEAD Working Paper No. 2019/25/EFE*.
Strang D, Macy MW. (2001). In search of excellence: Fads, success stories, and adaptive emulation. *American Journal of Sociology 10*(1), 147–182.
Suarez FF. (2005). Network effects revisited: The role of strong ties in technology selection. *Academy of Management Journal 48*(4), 710–720.
Teece DJ. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal 28*(13), 1319–1350.
Terlaak A, Gong Y. (2008). Vicarious learning and inferential accuracy in adoption processes.

Academy of Management Review 33(4), 846–868.
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.
Toole JL, Cha M, González MC. (2012). Modeling the adoption of innovations in the presence of geographic and media influences. *PLOS ONE 7*(1).
Tucker C, Zhang J. (2010). Growing two-sided networks by advertising the user base: A field experiment. *Marketing Science 29*(5), 805–814.
Valdez A. (2018), March 12. Inside the Vatican's first-ever hackathon. *Wired*.
Varian H, Shapiro C. (1998). *Information rules: A strategic guide to the network economy*. Harvard Business School Press: Boston.
Venkataraman V, Ceccagnoli M, Forman C. (2018). Multi-homing with platform ecosystems: The strategic role of human capital. *Georgia Tech Research Paper No. 18-8*. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3134846.
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.
Wade JB, Porac JF, Pollock TG, Graffin SD. (2006). The burden of celebrity: The impact of CEO certification contests on CEO pay and performance. *Academy of Management Journal 49*(4), 643–660.
Ter Wal A, Criscuolo P, Salter A. (2011). *Absorptive capacity at the individual level: An ambidexterity approach to external engagement*. DRUID Conference Paper on Innovation, Strategy, and Structure.
Watts DJ, Strogatz SHH. (1999). Collective dynamics of "small-world" networks. *Nature 393*(6684), 440–442.
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: Bingley, UK: 27–67.
Wu A. (2016). Organizational decision-making and information: Angel investments by venture capital partners. *Academy of Management Proceedings*. Available at: https://doi.org/10.5465/ambpp.2016.4.
Wu A, Clough DR, Kaletsky S. (2019). Nascent platform strategy: Overcoming the chicken-or-egg dilemma. *Harvard Business School Technical Note 719-507*.
Wu L, Wang D, Evans JA. (2018). Large teams develop and small teams disrupt science and technology. *Nature 566,* 378–382.
Yoffie DB, Casadesus-Masanell R, Mattu S. (2003). Wintel (A): Cooperation or conflict? *Harvard Business School Case Study No. 9-704-419*.
Yoo Y, Henfridsson O, Lyytinen K. (2010). The new organizing logic of digital innovation: An agenda for information systems research. *Information Systems Research 21*(4), 724–735.
Zhu F, Iansiti M. (2012). Entry into platform-based markets. *Strategic Management Journal 33*(1), 88–106.
Zittrain JL. (2006). The generative internet. *Harvard Law Review 119*(7), 1974–2040.

FIGURES

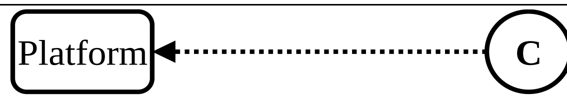
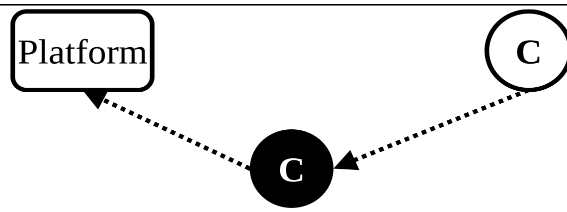
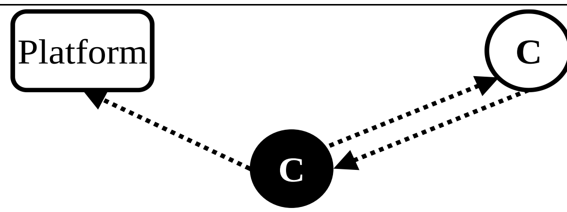
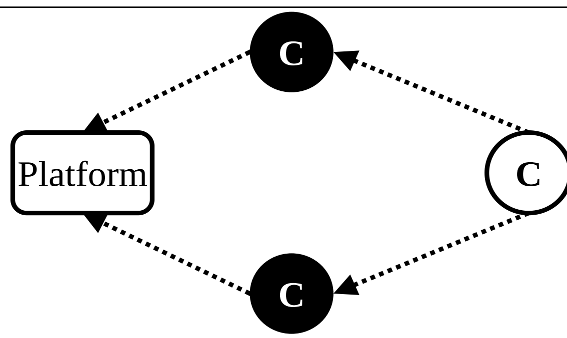
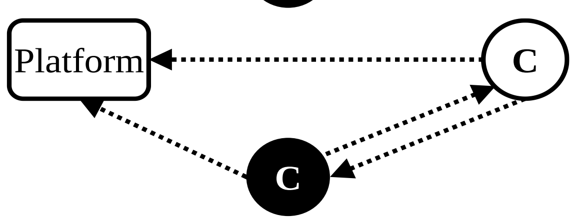
Diffusion Channel *Social Mechanism*	**Platform–Complementor Interaction(s)** Prior Adopter & Prospective Adopter
Sponsorship	
Awareness generation. *Economic inducement.*	
Social Learning	
Prior adoption observation. (**H1**) *Successful outcome observation.* (**H2**)	
Knowledge Exchange	
Technical knowledge exchange.	
Social Coordination	
Non-linear adoption. (**H3)**	
Sponsor amplification. (**H4**)	

Figure 1: Diffusion channels and social mechanisms at temporary gatherings.

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 and GitHub. *Expected Subsidy* is reported in thousands of USD.

Dependent Variables	**Obs.**	**Mean**	**Std. Dev.**	**Min.**	**Max.**
Platform Adoption	786,240	0.001	0.026	0	1
Main Independent Variables					
Hackathon Attendance	786,240	0.036	0.185	0	1
Local Adoption Rate	786,240	0.049	0.119	0	1
Winner Adoption	786,240	0.018	0.133	0	1
Control Variable					
Project Experience	786,240	8.160	11.415	0	100
Expected Subsidy (thousands)	786,240	0.005	0.121	0	15
Platform Stock	786,240	3.260	4.692	0	28

Table 2: Correlation matrix for independent variables.

Variable		**1**	**2**	**3**	**4**	**5**	**6**
Hackathon Attendance	**1**	1.000					
Local Adoption Rate	**2**	0.191	1.000				
Winner Adoption	**3**	0.154	0.299	1.000			
Expected Subsidy	**4**	0.203	0.048	0.024	1.000		
Project Experience	**5**	0.071	0.217	0.103	0.024	1.000	
Platform Stock	**6**	0.036	0.193	0.068	0.007	0.317	1.000

Table 3: Linear regression on *Platform Adoption*. In Models (3.0) through (3.4), we run linear probability models with a dependent variable of *Platform Adoption*. Across all models, the control variables *L Expected Subsidy*, *L Project Experience*, *L Platform Stock*, 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.

		(3.0)	**(3.1)**	**(3.2)**	**(3.3)**	**(3.4)**
		Platform Adoption				
	Hackathon Attendance	0.017	0.017	0.014	0.014	0.004
		(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
		[0.000]	[0.000]	[0.000]	[0.000]	[0.003]
H1	Local Adoption Rate		0.010	0.010		0.001
			(0.001)	(0.001)		(0.000)
			[0.000]	[0.000]		[0.000]
H2	Winner Adoption			0.006	0.006	0.004
				(0.002)	(0.002)	(0.002)
				[0.003]	[0.002]	[0.066]
H3	Local Adoption > 25%				0.001	
					(0.000)	
					[0.000]	
H3	Local Adoption > 50%				0.001	
					(0.000)	
					[0.006]	
H3	Local Adoption > 75%				0.007	
					(0.002)	
					[0.001]	
H4	Local Adoption Rate					0.075
	X Hackathon Attendance					(0.008)
						[0.000]
	L Expected Subsidy	0.015	0.014	0.015	0.016	0.009
		(0.008)	(0.008)	(0.008)	(0.009)	(0.007)
		[0.072]	[0.088]	[0.081]	[0.068]	[0.246]
	L Project Experience	-0.001	-0.001	-0.001	-0.001	-0.001
		(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
		[0.000]	[0.000]	[0.000]	[0.000]	[0.000]
	L Platform Stock	0.001	0.001	0.001	0.001	0.001
		(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
	Adjusted Within R^2	0.023	0.025	0.025	0.024	0.035
	Developers	1,302	1,302	1,302	1,302	1,302
	Observations	786,240	786,240	786,240	786,240	786,240

Table 4a: Linear regression on *Platform Adoption*. In Models (4a.1) through (4a.3), we run linear probability models with a dependent variable of *Platform Adoption* for sub-samples of developers. In these models, the control variables *L Expected Subsidy*, *L Project Experience*, *L Platform Stock*, and fixed effects for platform-month are included. We do not include individual-level fixed effects in any of the models. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	DV: Platform Adoption	**(4a.1)** Intermediate Group	**(4a.2)** Novice Group	**(4a.3)** Expert Group
	Hackathon Attendance	0.012	0.016	0.013
		(0.001)	(0.003)	(0.003)
		[0.000]	[0.000]	[0.000]
H1	Local Adoption Rate	0.008	0.010	0.015
		(0.001)	(0.002)	(0.004)
		[0.000]	[0.000]	[0.000]
H2	Winner Adoption	0.006	0.003	0.013
		(0.002)	(0.004)	(0.006)
		[0.011]	[0.506]	[0.017]
	Control Variables	YES	YES	YES
	Platform X Month FE	YES	YES	YES
	Adjusted Within R^2	0.021	0.022	0.042
	Developers	696	291	315
	Observations	420,063	175,797	190,340

Table 4b: Comparison of coefficients between technological experience. In Models (4b.1) and (4b.2) we compare the difference in the coefficients between our expert group and the novice and intermediate groups, respectively. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	(4b.1)	**(4b.2)**
	Difference	
	Expert vs. Novice Groups	Expert vs. Intermediate Groups
Hackathon Attendance	-0.003	0.002
	(0.005)	(0.004)
	[0.546]	[0.628]
Local Adoption Rate	0.005	0.007
	(0.004)	(0.004)
	[0.200]	[0.063]
Winner Adoption	0.011	0.007
	(0.007)	(0.006)
	[0.128]	[0.238]

Table 5: Social Tie Formation Analysis. Monthly Hazard Analysis for Developer Tie Formation. Columns 1, 2, and 3 use a Logit Specification; Columns 4, 5, and 6 use a Rare Events Logit Specification. Columns 1 and 4 are our baseline results that examine the monthly likelihood of two individuals forming ties. Columns 2 and 5 examine the yearly likelihood of tie formation. Columns 3 and 6 drop from the sample developer dyads that attend their first hackathon after 2016 to address potential missing data.

	Logit			Rare Events Logit		
	(5.1)	**(5.2)**	**(5.3)**	**(5.4)**	**(5.5)**	**(5.6)**
Variable	Baseline	Year	Drop Post-2016	Baseline	Year	Drop Post-2016
Hackathon Attendance	1.263	1.260	1.306	1.247	1.234	1.285
	(0.293)	(0.318)	(0.315)	(0.290)	(0.313)	(0.310)
	[0.000016]	[0.000076]	[0.000034]	[0.000016]	[0.000082]	[0.000035]
Observations	1,200,295	102,900	1,081,308	1,200,295	102,900	1,081,308

Table 6: Evidence for Social Diffusion Channels. This table summarizes the mechanisms we highlight, divided into channels of *Social Learning*, *Knowledge Exchange*, and *Social Coordination*. We summarize the empirical evidence base for each mechanism, which includes the corresponding measure used in the quantitative analysis and an illustrative quotation from an interviewee in our qualitative study.

Diffusion Channel Definition	**Evidence Base** Quantitative Measure & *Illustrative Qualitative Quotation*
Social Learning	
Frequency-Based Imitation Merely viewing others using a platform reduces prospective adopter's perceived uncertainty in the platform's capabilities, also known as an "informational cascade."	Positive relationship between Local Adoption Rate and hazard of platform adoption. (H1) *"If more people use it, it's a lower risk because it will be useable. If only one other team is using one API, how do you know if it's working or if this team was just lucky to get it running?"*
Outcome-Based Imitation When a developer uses the platform to win a prize (i.e., positive outcome), this draws attention to it. Prospective adopters raise their perception of the platform's value.	APIs used in projects that win a main hackathon prize are adopted by other participants at a higher rate. (H2) *"If it's a grand prize winner, that would be something like, wow, I want to do that too. So right after I get home, I look into the technology they used to build their app."*
Knowledge Exchange	
Platform users teach prospective adopters how to adopt the platform.	Experts are most affected by Local Adoption Rate due to having higher absorptive capacity. *"The best developers, the ones gunning for the prize, know that they can get whatever working by the end of the hackathon, so they would take more chances on a new technology which is hard because there is also less documentation they can find on Stack Overflow [developer Q&A forum]. But they also knew what questions they needed to ask when they hit roadblocks."*
Social Coordination	
Bandwagon Recognition Possible adopters recognize network effects matter and aim to converge on a dominant platform	Steep non-linearity of relationship between Local Adoption Rate and hazard of platform adoption. (H3) *"It's much more noticeable at a hackathon when people have confidence in a certain technology 'cause you will hear people talk about it, and you will see, like, lots of people talking to a certain company about their technology. And that makes it salient in a way that, like, no individual is really usually tracking, like the popularity of random technologies on the Internet. You're not doing that. You're only looking up a question when you have them about the thing that you care about in that moment."*
Sponsor Amplification Drawing on balance theory: interactions at a gathering establish a consensus sentiment. Platform's presence as sponsor helps make this sentiment positive.	Impact of Local Adoption Rate is stronger when platform owner is hackathon sponsor. (H4) *"A lot of people haven't really formed the ability to just look past the marketing or the inertia of a certain thing. This is why when new technologies come out, you just see so much inertia because everyone has a fear of missing out. This is what everyone is doing so, it's what I need to be doing."*

APPENDIX

This appendix documents details on the empirical context, data construction methodology, and additional empirical tests to supplement the main manuscript. The order of the items in this appendix follows the order in which they are mentioned in the main paper. First, we provide contextual background on hackathons in organizational research. We then describe the hackathon sample construction criteria and provide examples of included hackathons. Third, 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). Fourth, we describe the construction of the *Technological Expertise* variable. Fifth, to address selection by developers towards specific hackathons, we report a balance test and parallel trends test on pre-event observables, finding no evidence of selection on observable characteristics. Sixth, we verify our main results with an instrumental variables analysis based on the instrument of developer–hackathon geographic distance. Seventh and eighth, as tests to address potential selection by developers towards specific hackathons, we examine whether sponsor prominence and platform media coverage moderate *Hackathon Attendance*, and we find support that developers are not heavily influenced by these factors. Ninth, we examine the moderating role of hackathon duration; we find that longer hackathons have a stronger effect on developer behavior, supporting our theory which rests on social interactions as an important factor relating hackathon attendance to platform adoption. Tenth, we examine heterogeneity across different types of hackathons, and we find that our main results are robust in each of the different sub-samples. Finally, we conclude by describing the methodology used in our qualitative interview process with hackathon stakeholders.

Hackathons as an organizational context

To provide context for our theory development, we first document the hackathon phenomenon based on archival research and a program of 30 interviews with hackathon stakeholders.

Definition and history

A hackathon is an event that brings together participants who generate ideas or products over a short period of time. Past studies and reports emphasize the creative and collaborative approach to problem-solving that takes place at a hackathon (e.g., Arena et al., 2017; Lifshitz-Assaf, 2018). The tight time frame of one to several days creates sprint-like conditions. Most hackathons are, at least nominally, structured as a competition, with prize(s) presented at a public closing ceremony, where the winner(s) and runner(s)-up demonstrate their product to a broad audience. In our study, we focus on open hackathons for software developers with a physical venue for participants to co-locate.[24] Open hackathons invite members of the public to participate, enabling us to study platform adoption by the broader developer population.[25]

While the use of the term hackathon has evolved over time,[26] hackathons events of the format that we focus on date back to 1999, when developers in the open-source community came together to work on the operating system OpenBSD,[27] and Sun Microsystems challenged their JavaOne conference attendees to write an application for the Palm device (Aviram, 1999). In the late 2000s, hackathons began to grow in popularity, taking place at trade conferences like TechCrunch Disrupt. In 2011, the Wall Street Journal (Glazer, 2011) noted that "Hackathons,

[24] Hackathons that take place entirely online and events (sometimes referred to as hackathons) that focus on other creative pursuits, such as developing a business plan or writing music, lie outside the scope of this study.
[25] In contrast to open hackathons, closed company hackathons limit their attendees to members of an organization or firm, e.g., Microsoft and Facebook run internal hackathons for their own employees to develop projects, of their own choosing, for the company.
[26] The Oxford English Dictionary first records the word "hackathon" in use in 1985 to mean any event involving physical hacking, as in the action of weapons in a close-fought military battle. The word first shows up with regard to computing in a Usenet newsgroup in 1990 when it appears to refer to individual isolated development work.
[27] See, "OpenBSD Hackathons," http://www.openbsd.org/hackathons.html [9 January 2020].

once an obscure corner of the computer-programming world, are becoming more mainstream." Hackathons then made their way to universities. The first student-run university hackathon, PennApps, took place at the University of Pennsylvania in fall 2009 with 17 teams; PennApps ran its 20th semi-annual hackathon in fall 2019. And as of 2020, hackathons take place at all major U.S. and Canadian universities and many universities globally.

Hackathons involve three main types of stakeholders. First, an organizer operates the hackathon and brings in the other stakeholders. Second, a sponsor is a third party, usually a for-profit firm, providing financial and in-kind support for the operations of the hackathon. Third, a developer participates by developing software at the hackathon; a developer is a prospective complementor for the sponsors who own platform ecosystems.

Organizers

Hackathon organizers consist of industry trade groups (e.g., Python Software Foundation),[28] universities and student groups (e.g., TreeHacks at Stanford University), governmental entities (e.g., U.S. General Services Administration),[29] non-profit interest groups (e.g., Roman Catholic Church; Valdez, 2018), and for-profit firms. The organizer advertises the event to potential developer attendees and sponsors who provide resources needed to operate the event. The organizer addresses the needs of developers—including food, overnight accommodations, work space, and Wi-Fi—so developers can focus on work.[30] The organizer schedules both social and professional development activities at the hackathon. The organizer determines, through volunteer judges they recruit, the winner of the generic grand prize that all developers are eligible for. The organizer generally holds the hackathon at their own office or campus or

[28] See, "About PyCon," https://us.pycon.org/2019/about/ [9 January 2020].
[29] See, "GSA customer experience hackathon 2019," https://digital.gov/event/2019/06/19/gsa-customer-experience-cx-hackathon/ [9 January 2020].
[30] See, "Hacking it out: Factual at Pennapps", https://www.factual.com/blog/hacking-it-out-factual-at-pennapps/ [9 January 2020].

sometimes at a larger venue: PennApps grew rapidly over the years and eventually moved from the University of Pennsylvania campus to the Philadelphia 76ers' professional basketball arena to accommodate the thousands of attendees.

In nearly all the examples that we study, hackathon organizers are not directly compensated for organizing the event. Our interviews suggest that most organizers operate hackathons because they enjoy building and leading a software development community. Trade groups or other special-interest organizers sometimes pursue an objective, such as promoting open-source technology (e.g., Python and PyCon), improving public services (e.g., U.S. GSA Customer Experience hackathon), or having social impact (e.g., Roman Catholic Church and VHacks). Regardless, hackathons typically remain broad in what they allow developers to build.

Sponsors

Hackathons are sponsored by third-party firms—both large incumbents (e.g., Microsoft) and smaller startups (e.g., Twilio)—that use the event to interact with attending developers. Most hackathon sponsors own one or more technology platforms, and our quantitative empirical analyses focus on platform owners that sponsor hackathons.

A sponsor agrees upon a sponsorship package with the organizer, who defines what access and interaction the sponsor has with developers at the hackathon. Organizers generally offer different tiers of sponsorship, between a few hundred dollars up to tens of thousands of dollars. At minimum, a sponsor can display its branding at the event, e.g., brochure, website, and banners. Nearly all sponsors provide developers with free promotional items, known as "swag," such as t-shirts or stickers. Sponsors gain the right to send company representatives to the hackathon, which consist largely of professional engineers employed by the sponsor. Many sponsors also provide developers with free development tools to lower the cost of adopting their technology; for example, Amazon Web Services (AWS) provides free credits for their cloud service, and Oculus provides their virtual reality hardware for developers to work with. On top of that, the sponsor can have a custom prize named after itself, where the sponsor determines the award criteria and provides the reward over and above the fee it pays to be a sponsor. In most cases, representatives from the sponsor do the judging for these awards.

As a key part of their strategy, most of these sponsoring firms prioritize activities to promote their platform among developers. This strategy and practice of building support for a platform technology is known in the industry as "evangelism," a term which hints at the seemingly religious nature of the platform–developer relationship. Some professional engineers specialize in platform evangelism and may hold the job title of "evangelist." These engineers serve as mentors at the hackathon, where they may staff a branded company table and roam around the event. While the sponsor sends mentors to promote itself and its technology, the mentors spend their time at the event educating and helping developers with anything in software development, whether related to their employer or not. At higher-priced sponsorship tiers, mentors get reserved time and space to lecture educationally on the sponsor's technology.

Sponsors affiliate with hackathons to market their overall brand, recruit employees, and promote the use of their technology platform by raising awareness among developers and directly training developers on it.

Developers

Software developers participate in the hackathon and are eligible to win prize(s). Most developers attend hackathons for free and do not pay a registration fee. Most developers are relatively young (approximately 18–35 years old) and are a mix of university students, prospective entrepreneurs, and employed adults, depending on the hackathon. Most developers attend local hackathons near where they live, although some fly to distant hackathons.

Developers form teams, either before or at the start of the hackathon. Teams select an application idea and during the hackathon they write and debug software code to build the application. Most developers spend nearly all their waking hours working, eating, or playing at the hackathon venue. Developers work with their teammates at tables in close proximity to other teams. Developers socialize informally throughout the hackathon. For hackathons that last longer than a day, the developers sleep in local accommodations or even on site. For example, developers at HackMIT slept on 500 air mattresses purchased by the organizer specifically for the event.[31]

In our interviews, we document that the vast majority of developers attend because they see it as a fun activity while doing something productive, even incidentally. Given the relatively small size of the more pecuniary incentives (prizes, food, promotional items), it appears that non-pecuniary incentives, at least in the short term, drive participation.

Distinction between hackathons and innovation contests

At face value, a hackathon may seem similar to an open innovation contest, a time-delimited competition that orients the efforts of a broad set of contributors toward the goals of the contest organizer. An extensive literature on open innovation contests documents how these contests incentivize participants to engage in creative problem-solving for the benefit of the organizer on a specific problem determined by the organizer (Boudreau, Lacetera, & Lakhani, 2011; Scotchmer, 2004; Terwiesch & Xu, 2008). These studies focus largely on the distribution of quality among participant contributions (Jeppesen & Lakhani, 2010; Terwiesch & Xu, 2008), with a core finding that a broader set of contributors leads to a lower average quality but higher

[31] See, "The Hackathon budget," https://medium.com/hackers-and-hacking/the-hackathon-budget-d636b5b2ed6c [23 June 2014].

maximum quality of contributions (Girotra, Terwiesch, & Ulrich, 2010; Terwiesch & Ulrich, 2009).

However, from the perspective of the sponsor, there are several key differences between hackathons and what has been addressed in this past literature on open innovation contests. First, we focus on physical hackathons with co-located participants, in contrast to the online contests studied most by the innovation contest literature. The physical nature of these hackathons allows for the in-person socialization and learning among developers that is essential to our theoretical contribution and the practical implications of when and why platform firms should leverage a hackathon for their own goals. These physical hackathons occur at a lower scale than online contests, given the higher variable cost of supporting a developer, e.g., food, work space, etc. However, this limitation on size implies that the engagement of a developer with the other developers, sponsors, and organizers is likely larger than for an online contest, which generally draws an order of magnitude more participants. Second, hackathons allow developers to work on a broader scope of projects, which they themselves own. In contrast to innovation contests that specify a problem that the organizer wants solved, hackathons provide few to no bounds on the projects that participants can work on. Even those with a particular interest, such as MIT's Hacking Medicine's Grand Hackfest which focuses on health care problems, do not state a specific problem or solution for developers to work on, other than the limitation that projects should address something in the realm of the topic of interest (e.g., health care; Marcus, 2014). Regardless of the exact requirements, of which there are few, the developers own all the intellectual property they generate. In innovation contests, the organizer takes ownership or reserves the right to use the output, which was the organizer's whole incentive in the first place. Third, and most importantly, developers at hackathons are largely motivated for their own enjoyment and not for short-term financial reasons. Some developers do aspire to win a grand prize or any of the sponsor-specific prizes. However, our interviews suggest that the possibility of a prize is a second-order consideration for most participants, and most developers do not even seek to win prizes.

Hackathons do serve as an opportunity for developers to create innovations of their own interest. Developers leverage the hackathon as a forcing mechanism to get themselves to build something that they otherwise would not be able to find time to do. Given the limited time and somewhat competitive nature of the hackathon, the hackathon serves as a way for developers to timebox their work and force themselves to get something done. Moreover, developers are also incentivized to work hard through peer pressure, either directly from their teammates or indirectly by being surrounded by other teams hard at work. A noted subset of developers also has entrepreneurial intentions: they use the hackathon as an opportunity to build out an idea for a software product and even to form a team. Hackathons do not take ownership of the intellectual property generated by developers, allowing them to go on to pursue for-profit ventures built on their hackathon work.

In summary, hackathons distinguish themselves from open innovation contests in that they allow developers to creatively develop products they themselves choose and own, while immersed in a physical context with ripe opportunity for in-person interaction that lends itself to learning.

Hackathon sample construction

Before getting into the specifics of the sample of hackathons used in the main study, we first provide a descriptive illustration of hackathons over time. Figure B1 shows the rapid growth of hackathons over time, divided by category. This figure includes a broader sample of hackathons than those we use in the main study. We revisit all hackathons listed on Devpost.com, our data source for hackathon occurrences. For each hackathon, we code the dates for each event and plot a histogram of hackathons over time. The histogram suggests that hackathons are a recent phenomenon, growing quickly in popularity between 2014 and 2016. The number of hackathon events that occurred in 2016 is more than three times greater than in 2014. There were very few hackathons of the type we study prior to January 2014, suggesting that our set of developers were unlikely to attend a hackathon before January 2014.

---------- INSERT FIGURE B1 ----------

Table B1 provides descriptive information on the construction of the sample of hackathons included in the study. The starting sample includes all hackathons appearing on 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 events in our study.

---------- INSERT TABLE B1 ----------

Table B2 shows an example of selected hackathon data available through Devpost.com used in the sample construction process. We filter the set of hackathons based on the name and physical event venue location. We exclude a hackathon if the hackathon name includes any mention of sponsoring platform(s) 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 B2 ----------

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 hackathons. For each platform, we scrape data from Programmable Web, which is one of the largest online repositories of platform APIs. Table C1 presents the list of APIs used in our study and the associated platform owner, category for each API, and the intial release date of the 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 C1 ----------

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 (e.g., "import ibmwatson") is different from invoking an API through a web service (e.g., "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 cannot 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 infer 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 C2 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 measure of *Platform Adoption* introduced in this paper.

---------- INSERT TABLE C2 ----------

Independent variable construction: *Technological Expertise*

We examine developer heterogeneity by categorizing developers based on their technological expertise. Among the broad set of developers attending hackathons, differences in human capital may allow some developers to adopt a platform more easily than others. After collecting data for thirteen distinct input variables reflecting developer characteristics, we apply latent class analysis (LCA) to classify our developers into mutually exclusive categories: *Novice*, *Intermediate*, and *Expert*.

Input data collection and variables

For each developer in the sample, we collect data on technical skills, formal education, and practical experience from GitHub and LinkedIn to get as comprehensive a sense of their technical experience as one can observe. For all input measures, we take their value on the date at the end of the time window for which we observe developers in our main analysis, i.e., as of November 2017. We do this so each developer falls into a specific category that does not change over time. With the data from Github used in the main analysis, we construct measures for the count of technology platform *Tools*, software projects *Projects*, and hackathons *Hackathons* that are attended by our developers. *Tools* and *Projects* resemble the time-variant versions of these variables in the main analysis (*Platform Stock* and *Project Experience*, respectively), except that they are fixed at a single point in time. In the main paper, we choose to focus on the first sponsored hackathon attended by a developer, so *Hackathon Attendance* can only be a maximum value of 1. For this analysis, we allow *Hackathons* to be greater than one if they attend more than one hackathon by the end of the observed time window.

We then combine this data with additional data from LinkedIn. We scrape the LinkedIn profile for each developer. Because LinkedIn provides dates for the work and educational experience of each developer, the data provides insight into the experience that a developer accumulates over time. *United States* indicates whether a developer resides in the United States, and zero otherwise. *Engineering Degree* indicates whether the developer holds a post-secondary degree, like a college degree, in an engineering field. *Graduate Degree* indicates whether the developer holds a graduate degree in any field. *PhD Degree* indicates whether the developer holds a PhD in any field. *Degrees* counts the number of post-secondary degrees held by a developer. *Education Duration* reflects the total number of years of post-secondary education that a developer received.

Startup Experience indicates whether the developer ever held a full-time position at an early-stage company, which we define as companies that are less than ten years old. *Technical Experience* reflects the number of years of experience the developer has had in a full-time, technical position. *Technical Positions* counts the total number of distinct technical internships and full-time positions held by the developer. *Skills* counts the number of self-reported "skills" on each developer's LinkedIn profile, which includes skill areas like UX design and cloud computing.

In summary, our observable developer characteristics consist of eight count variables (*Skills, Tools, Projects, Degrees, Hackathons*, *Education Duration*, *Technical Positions*, and *Technical Experience*) and five indicator variables (*United States*, *Engineering Degree*, *PhD Degree, Graduate Degree*, *Startup Experience*).

Classifying with latent class analysis

We use the LCA method to classify each developer into one of three mutually exclusive classes—which we refer to as *Novice*, *Intermediate*, and *Expert*—based on the observable developer characteristics described above. As an equivalent method to factor analysis, the LCA method takes the observable input variables to generate a probability of class membership for each observation, rather than a continuous measure (Rawlings & Friedkin, 2017). LCA fits a pre-specified number of unobserved classes to the data. The LCA method allows us to make predictions of the latent (unobserved) technological expertise of a developer based on the observable input variables that we collect from LinkedIn and GitHub.

Given that LCA assigns observations to a predetermined number of classes, we need to fix the number of classes in advance. For our preferred specification, we allow three classes of developers. To arrive at three classes, we test models with two, three and four developer classes. Across these models, we assess the goodness-of-fit of each model using Akaike's Information Criterion (AIC; Akaike, 1987) and Bayesian Information Criterion (BIC; Schwarz, 1978). We find the three-class model to be most representative.

In the preferred three-class model, we find noticeable differences across each of the three classes in the margins for each observable variable. Figure D1 depicts the predicted values for each observable developer characteristic across the three classes of developers. We find that *Novice* developers are low across all characteristics: they are relatively unskilled, lack formal education, and have little software development experience. *Intermediate* developers are skilled and well-educated (i.e., they have a high number of self-reported skills, advanced degrees and many years of formal education), but are inexperienced with software development (i.e., they have lower technical job experience and experience on GitHub). *Expert* developers are as skilled and well-educated as the intermediate group, but have stronger programming experience.

---------- INSERT FIGURE D1 ----------

Developer-platform pre-hackathon observables

To address the concern that developers are self-selecting into hackathons based on sponsor, we check for balance on pre-hackathon platform adoption and ongoing usage between (a) developers who eventually attend a hackathon sponsored by a focal platform and (b) developers who eventually attend an event that is not sponsored by that platform. We find no statistically significant differences between the two groups across various platforms. We present a portion of the analysis in Table E1 for three platforms: Amazon Web Services, Google, and Microsoft. These platforms are popular and frequently sponsor hackathons, and thus they are most susceptible to developer selection. For each platform, we find that the level of *Platform Adoption* and *Platform Development* is comparable between the groups in the period before the event. *Platform Development* counts a developer's active GitHub projects that use a specific platform. This variable captures not just new adoption but also the ongoing activity of developers who already adopted the platform. We also find no statistically significant differences between the two groups across all the platforms considered in our study. Finally, we perform an analysis that examines pre-event 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 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 E1 ----------

We also assess a visual representation of the effect of the main independent variables to identify any possible non-parallel trends, shown in Figure E1. We find no evidence of non-parallel trends. 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 FIGURE E1 ----------

Developer geographic proximity 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 the time 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 affected 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.[32] 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–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-event 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 Project 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 F1a 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 F1a, the *F* statistic for the instrument is 1026.3 for *Hackathon Attendance* and 534.8 for *Local Adoption Rate*. The coefficients for our estimates

[32] 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. See Picard, R. (2010).

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 F1a ----------

Table F1b 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 F1b.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 F1b.2, we examine *Local Adoption Rate* and find that an increase of ten percentage points in the *Local Adoption Rate* of a given platform is associated with a 17.7 percentage point increase in the likelihood of adoption.

---------- INSERT TABLE F1b ----------

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.

Platform sponsorship-level analysis

We consider the possibility that hackathon participants are drawn to the hackathon by the largest sponsor and that they 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(s) 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 (the most prominent sponsor) than adoption of the minor sponsors' platforms. We argue that if there is significant selection in *Hackathon Attendance*, the estimated effects on hypothesized relationships should be less significant for less-prominent minor sponsoring platforms at each hackathon.

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 G1, Model G1.1 includes our major sponsor moderator in a specification similar to Model 3.0 in Table 3 of the main manuscript. This moderator allows us to measure heterogeneity in the effect size 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 G1 ----------

Platform media coverage analysis

To further address the possibility that there may be differences between sponsors, we examine differences between platforms that are prominently featured in the news and those that are not. To construct this measure, we gather information about the number of annual news mentions for each platform from Factiva. We use a number of keywords for each platform to reduce the risk of false positive news mentions being included in the count of news mentions. For each year, we identify sponsors that are above the median (high media coverage) and below the median (low media coverage). This allows us to account for different growth rates of platforms during our observation window and to construct sub-samples of high media coverage and low media coverage observations for each of our developer-platform-month observations.

We present the results of our robustness tests between low media and high media coverage sponsors in Table H1a. If developers select into hackathons that have well-known platform sponsors, we would not expect to find an association between *Hackathon Attendance* and adoption of a low media coverage platform. We do find that there is a positive and significant effect for both low and high media coverage. There are also no significant differences between low media and high media sponsors on the coefficients on the *Local Adoption Rate X Hackathon Attendance* term as shown in Table H1b. This provides additional support that sponsorship positively moderates the social learning effect for both low media coverage and high media coverage sponsors.

---------- INSERT TABLE H1a AND H1b ----------

Hackathon duration analysis

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 data 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 to 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 I1a. In Model I1a.1, we run a similar specification as in Model 3.4 in Table 3 of the main manuscript, replacing *Hackathon Attendance* with our alternative independent variables of *Hackathon Attendance: Short* and *Hackathon Attendance: Long*. Consistent with Hypothesis 4, the coefficients on the *Local Adoption Rate X Hackathon Attendance: Short* and the *Local Adoption Rate X Hackathon Attendance: Long* interaction terms are positive.

---------- INSERT TABLE I1a ----------

The Wald tests in Table I1b demonstrate that the effect of the interaction between *Local Adoption Rate* and attendance is larger for those who attend a longer hackathon as opposed to a shorter one. 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 hazard of adoption, 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 I1b ----------

Hackathon type analysis

We examine heterogeneity between different types of hackathons. We categorize the hackathons in the sample into categories of non-profit (public) hackathons, trade hackathons, and university hackathons, based on the identity of the hackathon organizer, i.e., a non-profit interest group or governmental entity, an industry trade group, or a university, respectively. To do so, we hired a team of research assistants to independently assign each hackathon to a category based on the text description and location of each event from Devpost. Table J1a presents the sub-sample estimates, and Table J1b presents the differences in coefficients between the different developer hackathons. We find no significant differences between the hackathon types for the coefficients on *Local Adoption Rate* and *Winner Adoption*. Although there may be differences across hackathon types, we still find evidence for the social mechanisms we theorize. Interestingly, we do note significant differences between public hackathons and university hackathons. This may be driven by the incentives of developers at these events: at university events, developers may be more inclined to experiment and learn about new technologies.

---------- INSERT TABLE J1a AND J1b ----------

Qualitative interview process

To better understand the empirical setting and to enrich our evidence base, we carried out a program of qualitative interviews with hackathon stakeholders. We conducted interviews with 30 different individuals between May 2019 and August 2019. These individuals were all based in the United States or Canada, and several had experience with hackathons in Asia, Europe, Africa, and Australia. These interviews took place in person and via phone call. We recorded all of these interviews in full and transcribed them. The average length of an interview was 55 minutes.

The interviews covered all the various stakeholders involved in hackathons: organizers, sponsors, and developers. Several interviewees fell into more than one of these categories. In addition, we interviewed community leaders who manage organizations that work with and support hundreds of hackathons. Collectively, our interviewees attended several hundred hackathons (at least), dating from the mid-2000s up to the present day.[33] Table K1 details the

[33] We do not have an exact number on the total number of hackathons attended by our interviews because interviewees often attended more hackathons than they could remember. Several interviewees attended over 100 hackathons each.

backgrounds and hackathon experience of our interviewees. Of the organizers, we interviewed those who operated hackathons for industry trade groups, universities, government, non-profit interest groups, and for-profit firms. Several of these organizers began as developers participating at hackathons before they moved into organizing. Of the sponsor interviewees, we interviewed both full-time engineers who volunteer part-time to serve as on-the-ground mentors for their employer, as well as those who hold the job title of evangelist and specialize full-time in promoting sponsor technology to developers. Some of those evangelists also had decision-making power for which hackathons would receive sponsorship from their firm. And of the developer interviewees, we engaged with developers who attended only one hackathon and several who attended over a dozen, including those who won prizes. Several of these developers participated in hackathons when they were undergraduate students and went on to work at major technology firms as engineers and evangelists, where they still return to hackathons as a mentor.

We benefited from the trust of our interviewees, as many interviewees were referred to us from previous interviewees in a snowball sampling approach. Given that our authorship team had experience both organizing and participating in hackathons, our subjects viewed us as someone with whom they could engage candidly while using technical terminology specific to hackathons and to software development.

---------- INSERT TABLE K1 ----------

APPENDIX REFERENCES

Akaike H. (1987). Factor analysis and AIC. *Psychometrika 52*(3), 317–332.

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.

Aviram M. (1999). JavaOne's palm-sized winner. *JavaWorld*.

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

Girotra K, Terwiesch C, Ulrich KT. (2010). Idea generation and the quality of the best idea. *Management Science 56*(4), 591–605.

Glazer E. (2011). On your mark, get set, hack! *Wall Street Journal*.

Jeppesen LB, Lakhani KR. (2010). Marginality and problem-solving effectiveness in broadcast search. *Organization Science 21*(5), 1016–1033.

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.

Marcus AD. (2014). 'Hackathons' aim to solve health care's ills. *Wall Street Journal*. Available at: https://www.wsj.com/articles/hackathons-aim-to-solve-health-cares-ills-1396654504.

Picard R. (2010). GEODIST: Stata module to compute geographical distances. *Statistical Software Components S457147*.

Schwarz G. (1978). Estimating the dimension of a model. *Annals of Statistics 6*(2), 461–464.

Scotchmer S. (2004). *Innovation and Incentives*. MIT Press: Cambridge.

Terwiesch C, Ulrich KT. (2009). *Innovation tournaments: Creating and selecting exceptional opportunities*. Harvard Business School Press: Cambridge, MA.

Terwiesch C, Xu Y. (2008). Innovation contests, open innovation, and multiagent problem solving. *Management Science 54*(9), 1529–1543.

Valdez A. (2018). Inside the Vatican's first-ever hackathon. *Wired*.

APPENDIX FIGURE

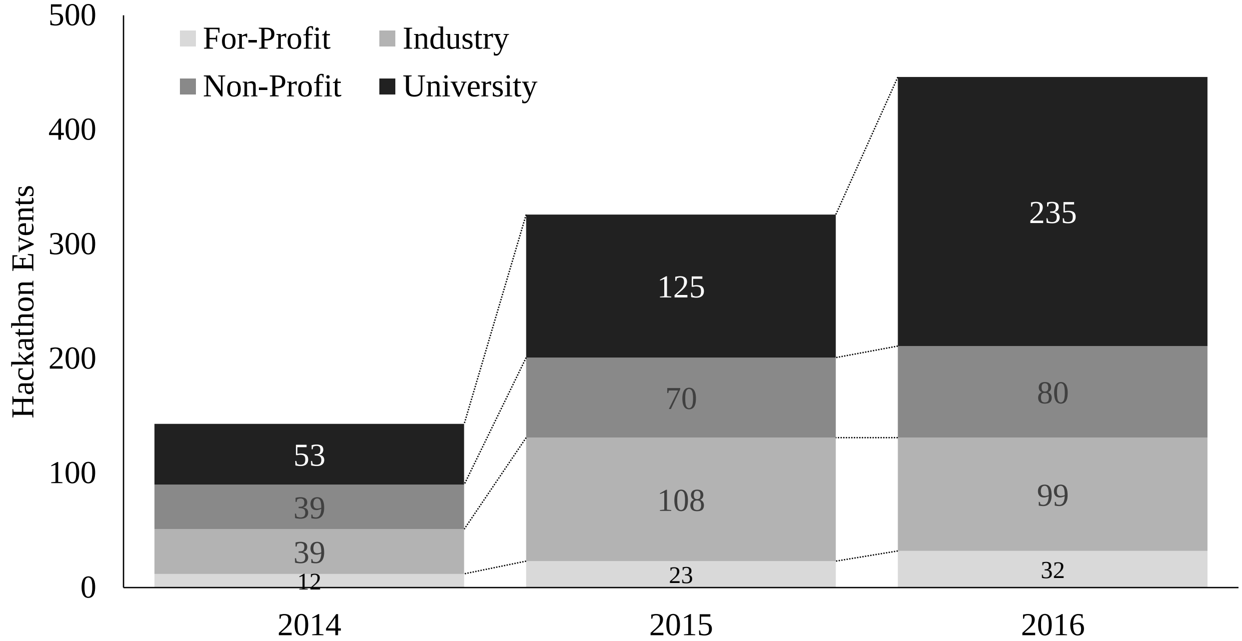


Figure B1: Count of hackathons over time by type of organizer.

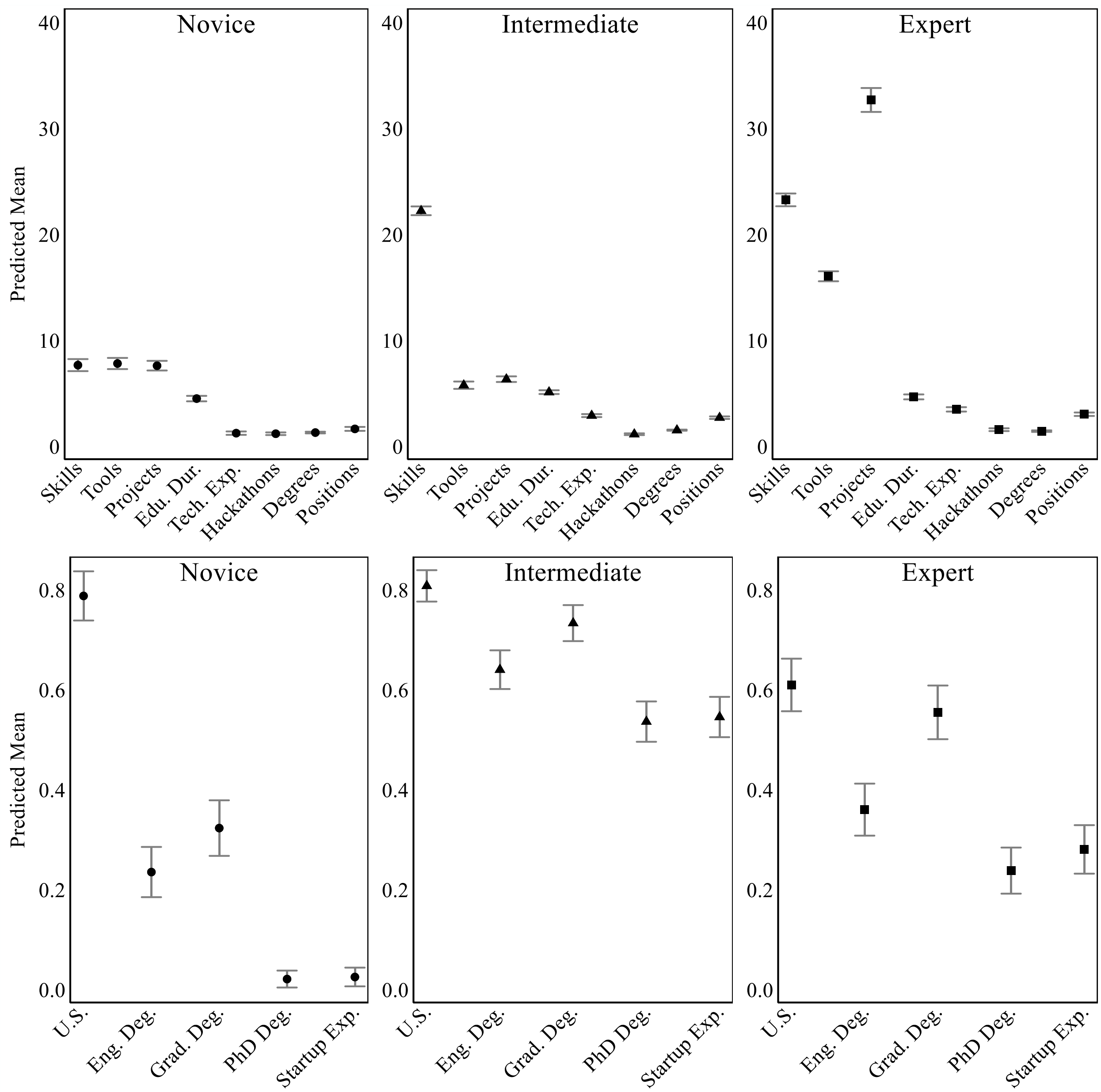


Figure D1: Margins plot for observable developer characteristics. This figure reports the predicted values of the developers in the Novice (left), Intermediate (center), and Expert (right) classes. The top panel shows predicted values for continuous variables. The bottom panel shows predicted values for indicator variables. Error bars represent 95% confidence intervals.

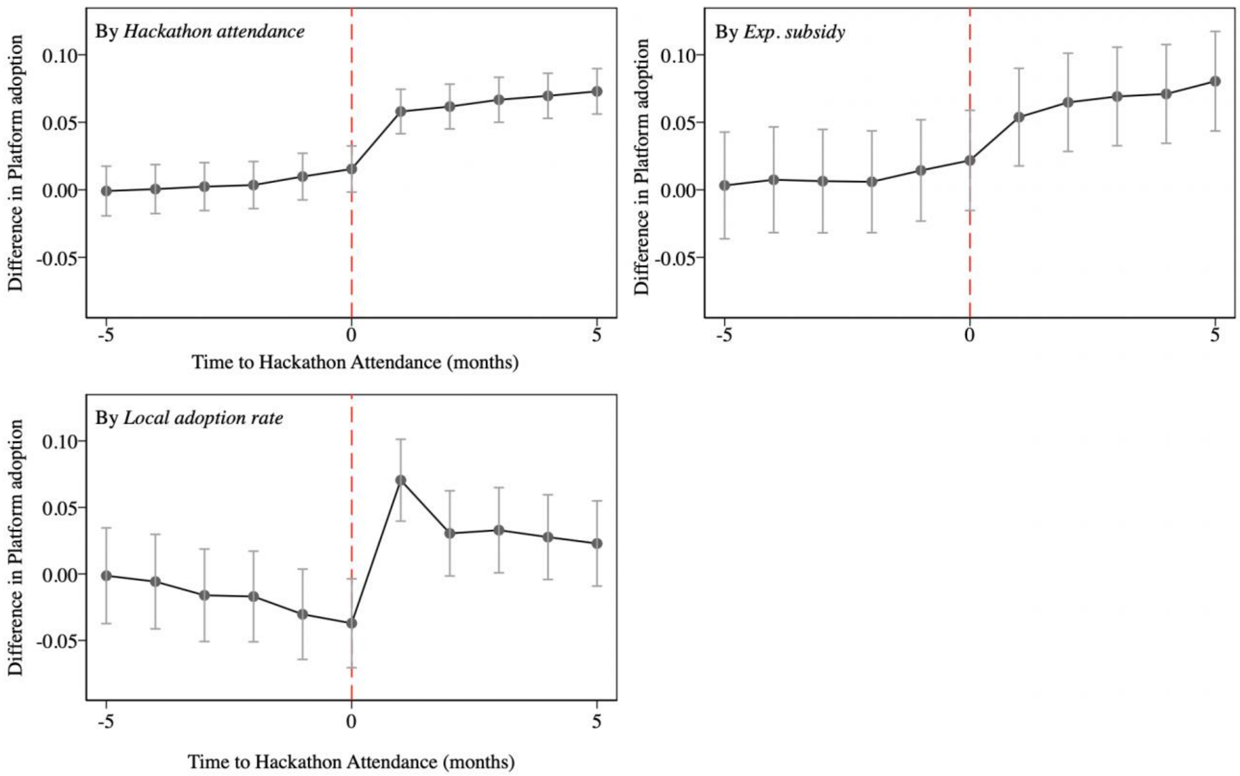


Figure E1: Pre- and post-hackathon trends. This figure depicts the differences in trends for Platform Adoption between treatment and control groups of developer–platform pairs. We compare: pairs that receive the *Hackathon attendance* treatment against those that do not (upper left); pairs that experience 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); and finally 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 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 TABLES

Table B1: 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 on 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 B2: 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 platform(s). 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 and *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, or iPad. *Microsoft Azure.* Microsoft will award a Surface Pro 4 and Surface Arc Mouse to each member of the team that best utilizes Microsoft Azure. *Yelp API.* Yelp will award a Leap Motion to each member of the team that best utilizes 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 C1: API list. For our 29 platforms, we search for the relevant API keyword on Programmable Web, and we 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. We also identify the year of the first version release of the API. 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	Founding Year
Amazon	alexa	Internet of Things	2012
Amazon	amazon	eCommerce	2009
Amazon	aws	Cloud	2008
Clarify	clarify	Audio	2014
Dropbox	dropbox	Storage	2010
Ebay	ebay	eCommerce	2011
Facebook	facebook	Social	2006
Facebook	fbsearch	Search	2010
Facebook	instagram	Photo	2012
Foursquare	foursquare	Social	2009
Google	firebase	Productivity	2012
Google	google	Search	2008
Google	nest	Internet of Things	2014
IBM	ibm	Cloud	2010
IBM	watson	Internet of Things	2014
Mastercard	mastercard	Payment	2012
Microsoft	azure	Cloud	2015
Microsoft	microsoft	Productivity	2009
Microsoft	outlook	Productivity	2015
MongoDB	mongodb	Productivity	2011
Paypal	paypal	Payment	2005
Spotify	spotify	Audio	2012
Staples	staples	eCommerce	2013
Twilio	twilio	Productivity	2009
Twitter	twitter	Social	2006
Uber	uber	Transportation	2014
Visa	visa	Payment	2014
Yahoo	yahoo	Search	2007
Yelp	yelp	Social	2007

Table C2: False positive verification example. We provide two examples of text from software code that contains a platform keyword ("google") and the results from the matching algorithm that we develop.

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

Table E1: Pre-event 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-event 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	1,072	0.532	230	0.522	0.011	0.766
Platform Development	1,072	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 F1a: First-stage coefficients. In this table, we show the first-stage results for the endogenous independent variables of *Hackathon Attendance* (F1a.1) and *Local Adoption Rate* (F1a.2). Our instrument for the analysis is *L Developer–Hackathon Distance*, the geodetic distance between each developer–hackathon. Across all models, we control for *L Project Experience* and include hackathon fixed effects and platform fixed effects. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	(F1a.1)	**(F1a.2)**
	Hackathon Attendance	Local Adoption Rate
L Developer–Hackathon Distance	0.023	0.006
	(0.001)	(0.000)
	[0.000]	[0.000]
L Project Experience	0.001	0.001
	(0.001)	(0.000)
	[0.008]	[0.000]
Platform FE	YES	YES
Hackathon FE	YES	YES
F Statistic	1,026.28	534.75
Observations	88,125	88,125

Table F1b: Reduced form coefficients on *Post-Event Platform Adoption*. In this table, we show the IV estimates from the second stage of the 2SLS model. The dependent variable is *Post-Event Platform Adoption*, which measures likelihood to adopt a platform at any point after the hackathon. Across all models, we control for *L Project Experience* and include hackathon fixed effects and platform fixed effects. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	(F1b.1)	**(F1b.2)**
	Post-Event Platform Adoption	
Hackathon Attendance	0.478	
	(0.047)	
	[0.000]	
Local Adoption Rate		1.771
		(0.179)
		[0.000]
L Project Experience	0.160	0.159
	(0.002)	(0.002)
	[0.000]	[0.000]
Platform FE	YES	YES
Hackathon FE	YES	YES
Adjusted Within R^2	0.056	0.050
Observations	88,125	88,125

Table G1: *Major Sponsor* moderator. We run a linear probability model with a dependent variable of *Platform Adoption*. In the model, the control variables *L Expected Subsidy*, *L Project Experience*, *L Platform Stock,* and fixed effects for platform-month are included. Variables preceded by L are logged as ln(1+x). Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	(G1.1)
Variable	Platform Adoption
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)
	[0.371]
L Expected Subsidy	0.016
	(0.008)
	[0.060]
L Project Experience	-0.001
	(0.000)
	[0.000]
L Platform Stock	0.001
	(0.000)
	[0.000]
Platform X Month FE	YES
Adjusted Within R^2	0.020
Developers	1,302
Observations	786,240

Table H1a: Platform media coverage. Across all models, we run linear probability models with a dependent variable of *Platform Adoption* for sub-samples of platform sponsors. In these models, the control variables *L Expected Subsidy*, *L Project Experience*, *L Platform Stock*, and fixed effects for platform-month are included. We do not include individual-level fixed effects in any of the models. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

		(H1a.1)	**(H1a.2)**
	DV: Platform Adoption	Low Media Coverage Sponsor	High Media Coverage Sponsor
	Hackathon Attendance	0.004	0.006
		(0.002)	(0.002)
		[0.088]	[0.000]
H1	Local Adoption Rate	0.000	0.002
		(0.000)	(0.000)
		[0.000]	[0.000]
H2	Winner Adoption	0.001	0.005
		(0.003)	(0.002)
		[0.762]	[0.040]
H5	Local Adoption Rate	0.065	0.074
	X Hackathon Attendance	(0.016)	(0.009)
		[0.000]	[0.000]
	Control Variables	YES	YES
	Platform X Month FE	YES	YES
	Adjusted Within R^2	0.024	0.038
	Developers	665	637
	Observations	401,902	384,338

Table H1b: Comparison of coefficients between sponsor media coverage. We compare the difference in the coefficients between low media and high media sponsors from Table H1a. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	(H1b.1) *Difference*
	Low Media Coverage Sponsor vs. High Media Coverage Sponsor
Hackathon Attendance	0.002
	(0.002)
	[0.362]
Local Adoption Rate	0.002
	(0.000)
	[0.000]
Winner Adoption	0.005
	(0.003)
	[0.139]
Local Adoption Rate	0.009
X Hackathon Attendance	(0.016)
	[0.577]

Table I1a: Hackathon duration moderators. We run a linear probability model with a dependent variable of *Platform Adoption*. We add independent variables for *Hackathon Attendance: Short* (event duration < 24 hours) and *Hackathon Attendance: Long* (event duration ≥ 24 hours). In the model, the control variables *L Expected Subsidy*, *L Project Experience*, *L Platform Stock*, and fixed effects for platform-month are included. Robust standard errors clustered by developer shown in parentheses. p-values shown in brackets.

		(I1a.1) Platform Adoption
	Attendance: Short	0.011
		(0.010)
		[0.251]
	Attendance: Long	0.015
		(0.011)
		[0.167]
H1	Local Adoption Rate	-0.003
		(0.001)
		[0.034]
H2	Winner Adoption	0.033
		(0.013)
		[0.013]
H5	Local Adoption Rate	0.258
	X Attendance: Short	(0.051)
		[0.000]
H5	Local Adoption Rate	0.399
	X Attendance: Long	(0.054)
		[0.000]
	Control Variables	YES
	Platform X Month FE	YES
	Developer FE	NO
	Adjusted Within R^2	0.026
	Developers	1,302
	Observations	786,240

Table I1b: Wald test comparing short and long hackathon duration estimates. This table provides Wald tests on the difference in estimated coefficients from Table I1a. We test the difference between groups 1 and 2 using the estimates from Model (I1a.1). Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

		(I1b.1) *Difference*
Group 1	Group 2	Platform Adoption
Attendance: Short	Attendance: Long	0.004
		(0.012)
		[.725]
Local Adoption Rate	Local Adoption Rate	0.141
X Attendance: Short	X Attendance: Long	(0.075)
		[.059]

Table J1a: Hackathon type. In Models (J1a.1) through (J1a.3), we run linear probability models with a dependent variable of *Platform Adoption* for sub-samples of hackathon events. In these models, the control variables *L Expected Subsidy*, *L Project Experience*, *L Platform Stock*, 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.

	DV: Platform Adoption	**(J1a.1)** Public Hackathon	**(J1a.2)** Trade Hackathon	**(J1a.3)** University Hackathon
	Hackathon Attendance	0.003	0.004	0.004
		(0.004)	(0.002)	(0.002)
		[0.249]	[0.056]	[0.046]
H1	Local Adoption Rate	0.001	0.001	0.001
		(0.001)	(0.000)	(0.000)
		[0.039]	[0.062]	[0.001]
H2	Winner Adoption	0.006	0.005	0.003
		(0.005)	(0.004)	(0.003)
		[0.178]	[0.179]	[0.228]
H5	Local Adoption Rate	0.038	0.057	0.088
	X Hackathon Attendance	(0.013)	(0.013)	(0.011)
		[0.003]	[0.000]	[0.000]
	Control Variables	YES	YES	YES
	Platform X Month FE	YES	YES	YES
	Adjusted Within R^2	0.054	0.021	0.039
	Developers	199	227	876
	Observations	119,825	137,343	529,058

Table J1b: Comparison of coefficients between hackathons. In Models (J1b.1) and (J1b.2) we compare the difference in the coefficients between public hackathons and trade and university hackathons, respectively, from Table J1a. Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	(J1b.1)	**(J1b.2)**
	Difference	
	Public vs. Trade Hackathons	Public vs. University Hackathons
Hackathon Attendance	0.001	0.001
	(0.004)	(0.004)
	[0.826]	[0.847]
Local Adoption Rate	0.000	0.001
	(0.001)	(0.001)
	[0.542]	[0.371]
Winner Adoption	-0.001	-0.003
	(0.006)	(0.006)
	[0.931]	[0.645]
Local Adoption Rate	0.018	0.050
X Hackathon Attendance	(0.019)	(0.017)
	[0.330]	[0.004]

Table K1: Interview subjects in qualitative study. *Role(s)* are the one or more roles that the interviewee held at past hackathons; for those with multiple roles, these different roles were at different hackathons and never at the same time in a single hackathon. *Initial* and *Final* reflect the first and most recent year in which the interviewee was involved in a hackathon. *Count* is the minimum number of hackathons we were able to validate that involved the interviewee; in several cases, the interviewee took part in too many hackathons to provide an exact number. *Current Job Title or Status* represents the interviewee's current position as of summer 2019 and not necessarily their position when they were previously involved in the hackathon(s).

Role(s)	Initial	Final	Count	Current Job Title or Status
Developer	2014	2019	4	UG Student in Computer Science
Developer	2015	2018	9	UG Student in Computer Science
Developer	2017	2019	2	Business Technology Analyst
Developer	2018	2019	7	UG Student in Computer Science
Developer	2018	2019	2	Senior Financial Analyst
Developer	2018	2018	3	UG Student in Computer Science
Developer, Mentor	2013	2019	6	Software Engineering Manager
Developer, Mentor	2014	2019	11	Software Engineer
Developer, Organizer	2011	2015	5	PhD Student in Computer Science
Developer, Organizer	2012	2016	4	Software Engineer
Developer, Organizer	2013	2016	15	Product Manager
Developer, Organizer	2015	2019	8	UG Student in Commerce
Developer, Organizer	2017	2019	5	Product Developer
Developer, Organizer	2018	2019	2	UG Student in Computer Science
Mentor	2006	2019	26	Developer Marketing
Mentor	2010	2019	100	Developer Advocate
Mentor	2011	2019	100	Developer Evangelist
Mentor	2013	2019	20	CTO & Founder
Mentor	2016	2019	7	Senior Software Engineer
Mentor	2018	2018	15	Startup Programs Leader
Organizer	2010	2019	20	Managing Partner
Organizer	2010	2012	8	Growth Engineering
Organizer	2011	2015	4	Product Manager
Organizer	2012	2014	3	Investment Associate
Organizer	2014	2019	5	Manager of Recruitment & Community
Organizer	2016	2019	5	UI/UX Visual Designer
Organizer	2017	2019	3	UG Student in Computer Science
Organizer	2019	2019	1	UG Student in Applied Mathematics
Community Leader	2007	2019	100	CEO & Founder
Community Leader	2009	2019	100	CEO & Founder