Iterative Coordination and Innovation *

Sourobh Ghosh †
Technology and Operations Management Unit
Harvard Business School

Andy Wu ‡
Strategy Unit
Harvard Business School

January 28, 2020

Abstract. Agile management practices from the software industry continue to transform the way organizations innovate across industries, yet they remain understudied in the organizations literature. We investigate the widespread Agile practice of *iterative coordination*: frequent meetings to coordinate individuals on innovation-oriented organizational goals. While the assumed purpose of iterative coordination is to generate innovation, there is limited empirical evidence as to whether this practice actually makes organizations more innovative. With the leading technology firm Google, we embed a field experiment within a hackathon software development competition to identify the effect of iterative coordination on innovation. We find that iterative coordination's influence on innovation is mixed: while iteratively coordinating firms develop more valuable products, these products are simultaneously less novel. This counterintuitive result highlights a paradox at the heart of iterative coordination—whereas it intends to promote innovation, applying it yields less novelty, a necessary component of innovation. Furthermore, by tracking software code, we find that iteratively coordinating firms favor integration at the cost of specialization to create new knowledge. A follow-on laboratory study documents that increasing the frequency and goal orientation of iterative coordination meetings reinforces value and integration, while reducing novelty and specialization. This article makes three key contributions: refining theoretical understanding of the relationship between goal coordination and innovation; introducing empirical methodology for generating data with hackathons and software code version-tracking; and responding to calls from the literature to study new methods of organizing and whether they offer new insights on organizational innovation.

Keywords: Innovation, Organizations, Goals, Integration, Specialization, Coordination, Field Experiment, Software Development

---

*The authors gratefully acknowledge feedback from Shiva Agarwal, Vikas Aggarwal, Daniel Albert, Ethan Bernstein, John Beshears, Afonso Almeida Costa, John Chen, David Clough, Johnathan Cromwell, Felipe Csaszar, David Hsu, Laura Huang, Rory McDonald, Cynthia Montgomery, Phanish Puranam, Dimo Ringov, Jan Rivkin, Stefan Thomke, and seminar participants at the Harvard Business School, 2018 Strategy Science Conference, 2018 Academy of Management Annual Meeting, 2018 Strategic Management Society Annual Conference, 2018 Academy of Management Special Conference Tel Aviv, and 2018 Association for Computing Machinery Conference on Human Factors in Computing Systems (ACM CHI). This research was made possible by support from the HBS Division of Research and Faculty Development, the National Science Foundation Graduate Research Fellowship, and Google LLC. Special thanks to Maxwell Bileschi of Google. Emily Chen, Jaeho Kim, Andrew Marder, and Jeff Strabone provided excellent research assistance. All opinions expressed herein are those of the authors only, and all errors are the responsibility of the authors.

†Address: Wyss House, 20 N Harvard St, Boston, MA 02163. Email: `sourobh@hbs.edu`.

‡Address: Morgan Hall 243, 15 Harvard Way, Boston, MA 02163. Telephone: 1 (617) 495-2795. Email: `awu@hbs.edu`.

1 Introduction

In a famous editorial, venture capitalist Marc Andreessen argued that "software is eating the world" (2011), predicting that the software industry would transform business. Software development management practices such as Agile are "eating" up traditional practices across industries and providing new ways to manage projects and design business models (Zott et al. 2011) and organizations (Burton et al. 2017). Agile has many prescribed implementations, though it is most closely associated with Scrum methodology and its eponymous practice of the "daily scrum" or stand-up meeting (Schwaber and Beedle 2001). These frequent meetings coordinate individuals on organizational goals, a practice we refer to as *iterative coordination*. Today, over 70% of organizations use iterative coordination to manage a wide array of software and non-software projects (Langley 2017).

Iterative coordination specifically intends to promote innovation. Given the increased rate of change in the environments in which organizations operate (MacCormack et al. 2001), scholars and practitioners note the need for organizations to continuously innovate (Brown and Eisenhardt 1997, Lee and Edmondson 2017). Here, speed in innovation is seen as a necessary response to the dynamic markets that cause less-innovative organizations to fail (Eisenhardt and Tabrizi 1995, Siggelkow and Rivkin 2005). In response, practitioners developed iterative coordination to rebut the extensive planning inherent to traditional coordination methods that they assume slow the speed and quality of innovation (Sutherland 2004, 2005). Jeff Sutherland, one of the original co-creators of Agile and iterative coordination as a formal practice,[1] notes with co-authors that, "Innovation is what agile is all about...Companies that create an environment in which agile flourishes find that teams can churn out innovations faster" (Rigby et al. 2016a). By rapidly iterating on organizational goals via iterative coordination, organizations can increase their speed and quality of innovation, out-performing their competitors (Vesey 1991).

Despite iterative coordination's explicit foundational focus on innovation, it is unclear what effect this practice has on actually generating innovation, particularly as we unpack innovation into its two necessary components: novelty, or the state of being new, and value, an innovation's economic worth (Amabile 1983, Singh and Fleming 2010, Kaplan and Vakili 2015). Especially when considering novelty, the practice of iterative coordination brings out conflicts in existing theoretical perspectives on the origins of innovation. The literature on creativity in organizations suggests that frequently iterating on organizational goals is necessary to achieve novel outcomes (Alexander and

Van Knippenberg 2014, Amabile and Pratt 2016, Cromwell et al. 2018). Given the challenges associated with pursuing truly novel ideas (Alexander and Van Knippenberg 2014), frequently revisiting goals is seen as an effective tool to motivate organizational members and keep them aligned in the pursuit of novelty (Shalley 1991, 1995, Madjar and Shalley 2008).

On the other hand, literature from the behavioral theory of the firm suggests that increased coordination on organizational goals drives out an important source of novelty in innovation—specialization in individual effort (Cyert and March 1963). Here, individual specialization can develop and expand the bounds of an organization's knowledge, helping organizations identify novelty (March and Simon 1958, March 1991, Taylor and Greve 2006). In contrast, coordination on goals facilitates integration (Okhuysen and Bechky 2009), which is assumed to trade off with specialization (Levinthal and Workiewicz 2018). Thus, it is assumed that coordinating on organizational goals will favor integration and inherently leave less time for the specialization necessary to identify novelty (Knudsen and Srikanth 2014). To adjudicate these competing views, this paper asks: What effect does iterative coordination have on innovation?

We partnered with Google LLC, a multinational information technology firm, to embed a field experiment within a public, one-day software application development competition, popularly known as a hackathon, to study the influence of iterative coordination on innovation. By randomly assigning treatment firms at the hackathon to regular stand-up meetings, we exogenously vary which firms engage in iterative coordination. This approach mitigates traditional endogeneity concerns associated with archival data approaches (Chatterji et al. 2016). To collect precise data tracking firms, we introduce a novel methodology leveraging the version-control systems used in software development. By documenting the progress of actual software code developed, we capture patterns of firm activities at a granular level over time (by minute) in a balanced panel dataset. Our partner provided performance assessments of each organization's final software applications. In addition, we run a follow-up experiment in the laboratory where we vary levers of managerial control for iterative coordination and further validate the internal consistency of our findings.

We find that iterative coordination's influence on innovation is mixed: while iteratively coordinating firms develop products that are more valuable, these products are simultaneously less novel. This striking result highlights a paradox at the heart of iterative coordination practice. Whereas iterative coordination intends to promote innovative outcomes, our findings demonstrate that applying such coordination yields less novelty, a necessary component of innovation. Furthermore, by tracking minute-by-minute changes in the software source code, we find that iteratively coordinating firms favor knowledge integration at the cost of in-depth, specialized knowledge creation by their members, in line with predictions from the behavioral theory of the firm that anticipate less novelty. In the follow-on laboratory study, we find that increasing the frequency and goal orientation of iterative coordination meetings reinforces integration and value, while reducing specialization and novelty in outcomes.

Our study offers three contributions to the organizations literature. First, we introduce iterative coordination as a managerially relevant practice to the organizations literature while broadening our theoretical understanding of the relationship between goal coordination and innovation. Our findings suggest that iterating on innovation-oriented goals paradoxically leads to less-novel outcomes, even in entrepreneurial settings and hackathons that prioritize innovation. Second, we make a number of methodological contributions to the empirical study of organizational innovation. We introduce hackathons as an externally valid research setting for organizations research, well-suited for experimental interventions such as ours, while producing real-world data on innovation process and outcomes. We also introduce software code tracking as a new data collection methodology for studying organizational innovation. Finally, we respond to recent calls from the literature to study new phenomena in organizing—and whether they offer new insights for innovation in organizations (Puranam et al. 2014, Burton et al. 2017). Despite the promise from practitioners that Agile balances the benefits of integration and specialization in organizing (Rigby et al. 2016b), we find that the foundational Agile practice of iterative coordination does not allow organizations to break this fundamental trade-off, ultimately favoring integration over specialization. Overall, we argue that managers often adopt iterative coordination for the wrong reasons: anticipating novel outcomes while ultimately receiving less novelty in practice.

2 Phenomenon of Iterative Coordination

2.1 Agile Methodology: Iterative Coordination in Practice

Agile methodology's foundational motive to increase speed in innovation continues to transform organizational practice (Rigby et al. 2016a). In February 2001, faced with the demands of creating robust software products for increasingly turbulent markets (Eisenhardt and Tabrizi 1995, MacCormack et al. 2001), 17 developers gathered in Snowbird, Utah to develop the principles behind what would become the Agile Manifesto. Seeking to rebut the extensive planning and linear approach of traditional manager-driven coordination methods, the Manifesto sought to promote speed in innovation via principles such as the continuous and frequent delivery of software to customers (Beck et al. 2001). To achieve this speed, the designers of Agile methodology reasoned that more frequent coordination on organizational goals was necessary (Cao and Ramesh 2007, Furr et al. 2016). To this end, existing methods for organizational coordination, such as traditional planning and role assignment by managers (Van de Ven et al. 1976, Okhuysen and Bechky 2009), were deemed ineffective (Hoppmann et al. 2019, Lu et al. 2019). With new methods to rapidly iterate on organizational goals, Agile methodology could make firms be more flexible and adaptive to their environments, helping them achieve product-market fit (Ries 2011). More importantly, greater speed in innovation could help firms out-innovate to achieve competitive advantage (Vesey 1991).

While Agile and its descendent frameworks prescribe a variety of practices to facilitate speed of innovation with frequent goal coordination, the use of iterative coordination remains a unifying factor across these diverse implementations. Iterative coordination, or frequent meetings to coordinate individuals on organizational goals, is best embodied in the widespread use of frequent stand-up meetings. These stand-up meetings give members of an organization the opportunity to iterate on a shared organizational goal. The leading Agile framework, Scrum, is even eponymously named after these "scrum" stand-up meetings, illustrating their importance in Agile organizing.

The practice of iterative coordination consists of frequent meetings to iterate on organizational goals. Most organizations adopt specific discussion questions to be repeated across all of these meetings. In a common formulation—which we apply in our experimental studies—each meeting consists of discussion around three questions: (1) "What have you accomplished since the last meeting?"; (2) "What are your goals until the next meeting?"; and (3) "What are your goals for the end of the project (and have they changed)?"[2] Question 1 updates members of an organization about prior work towards a pre-existing organizational goal. Question 2 divides and allocates tasks in the pursuit of a pre-existing organizational goal. Question 3 prompts members of an organization to revisit and update their shared organizational goal. Question 3 especially differentiates iterative coordination from a regular meeting by forcing the organization to update its high-level organizational goals. Through this third question, organizations could more frequently iterate on and adapt shared goals to be more innovative.

In the absence of iterative coordination, how would an organization and its members behave, all other factors held equal (e.g., organizational structure, resources, etc.)? An extensive body of prior work documents how members of an organization have limited impetus to discuss their shared organizational goal. Should members of an organization choose to meet, they may update each other on prior tasks and decide which tasks to complete until the next meeting—but it is less likely that a simple meeting in and of itself would prompt iterating on organizational goals. First, the mentally taxing cognition needed for goal-setting does not represent the natural disposition of most workers (Critcher and Ferguson 2016, Pervin 1992, Seijts et al. 2004). Second, members in an organization naturally seek to avoid conflict, an unavoidable consequence when discussing organizational goals (Cyert and March 1963, Gavetti et al. 2012). Consequently, without a forcing mechanism for such conflict as presented by iterative coordination, there is less tendency to revisit higher-level shared goals. Finally, empirical evidence from group work studies suggests that when groups try to coordinate without a formal intervention (such as iterative coordination), the group integrates less knowledge from its members (Okhuysen and Eisenhardt 2002). This knowledge integration is necessary to revisit organizational goals (Okhuysen and Bechky 2009). In summary, in the absence of iterative coordination, organizations update their shared goals less frequently.

2.2 Iterative Coordination and Innovation

Before examining iterative coordination's potential effect on innovation, it is helpful to break down innovation into its components of novelty and value (Amabile 1983, Singh and Fleming 2010, Kaplan and Vakili 2015). Recent work in adjacent literatures highlights the benefit of breaking down innovation into its constituent components. In particular, separating out novelty from value in innovation helps expose gaps and inconsistencies in prior theorizing. For instance, Kaplan and Vakili (2015) and Jung and Lee (2016) both demonstrate that while specific mechanisms in their respective literatures predict innovation as high-level outcome, these very mechanisms fail to predict novelty.[3] Given our interest in iterative coordination's effect on innovation, we are especially motivated by understanding its impact on novelty, which is central to any conceptualization of innovation.

The literature on creativity in organizations suggests that frequently iterating on organizational goals is necessary to achieve novel outcomes (Alexander and Van Knippenberg 2014, Amabile and Pratt 2016, Cromwell et al. 2018). In particular, the process of revisiting and iterating organizational goals can reinforce individual motivation, which is an important antecedent to creative and innovative outcomes (Amabile and Pratt 2016). Evidence for the power of goals on creativity emerges across multiple levels of analysis. For instance, individuals assigned to creative goals yield more-novel output than those who are not assigned to these goals (Shalley 1991, 1995, Madjar and Shalley 2008). At a team level, Gilson and Shalley (2004) find that teams reporting shared goals engaged more in the creative process. Especially for more dynamic tasks such as new product development, a process for iterating on goals can help realign organizational members after inevitable failures that occur when pursuing novel, albeit risky, ideas (Alexander and Van Knippenberg 2014). In sum, frequently iterating on organizational goals is seen as an effective tool to help keep organizations on track in the pursuit of novelty.

On the other hand, literature from the behavioral theory of the firm suggests that increased coordination on organizational goals drives out an important source of novelty and innovation—specialization of individual effort (Cyert and March 1963). Here, individual specialization develops and expands the bounds of an organization's knowledge, helping organizations identify novelty (March and Simon 1958, March 1991, Taylor and Greve 2006). In contrast, coordination on goals facilitates integration (Okhuysen and Bechky 2009), which is assumed to trade off with specialization (Levinthal and Workiewicz 2018). In light of the trade-off between integration and specialization in organizations, it is assumed that coordinating on organizational goals will favor integration and inherently leave less time for the specialization necessary to generate novelty (Knudsen and Srikanth 2014). Nonetheless, existing studies in this tradition generally overlook how trading off between coordination and specialization affects the distinct components of innovation, i.e., value and novelty. Instead, this literature focuses on how the trade-off informs firm performance in general. For instance, many studies demonstrate that effective coordination among specialists enables the organization to identify high-value ideas (Rivkin and Siggelkow 2003, Knudsen and Srikanth 2014, Levinthal and Workiewicz 2018). Although value is clearly an important dimension of performance, a valuable idea does not necessarily have to be novel (Kaplan and Vakili 2015).

3 Primary Study: Software Development Field Experiment

3.1 Experimental Setting

To facilitate causal empirical identification of the theorized effects of iterative coordination, we design and deploy a field experiment. Given iterative coordination's roots in the software industry (Rigby et al. 2016b), we focus on the context of managing software development. To maintain managerial relevance, we sought an externally valid experimental context to demonstrate the impact of iterative coordination as a managerially implementable practice. We begin by presenting background on the externally valid empirical context—a software development competition, known generally as a hackathon—followed by the exposition of our experimental procedures leveraging the widespread managerial phenomenon of Agile and Scrum stand-up meetings as our experimental treatment of iterative coordination.

We partnered with Google LLC (Google), a multinational information technology firm, to embed a field experiment within a one-day software application ("app") development competition, or hackathon, hosted on the campus of a university in the northeastern United States.[4]

3.1.1 Hackathons: Background and External Validity Over the last decade, hackathons emerged to play a pivotal role in software development culture and practice (Broussard 2015, Leckart 2015). Hackathons commonly entail sets of software developers who compete in a contest to develop and present working software by the end of a timeframe of a day or two (Leckart 2012). While some hackathons focus on particular themes or interest areas, they generally operate as open-ended design contests which embrace ill-structured problems (Simon 1973), such as those encountered in early-stage strategy and entrepreneurial settings (Ott et al. 2017). For instance, a charge to action at a hackathon could be "create a mobile app that achieves X," where X is an under-specified, ill-articulated latent consumer need (Von Hippel 2005).

The competition aspect of the hackathon typifies the dynamic and entrepreneurial settings in the software industry where firms implement iterative coordination in practice. Each set of participants mirrors the composition of an archetypal software start-up firm in terms of skills and size. In fact, many successful startup firms began as hackathon projects: the popular messaging app GroupMe was conceived at the 2010 TechCrunch Disrupt hackathon and acquired a year later by Skype for about $80 million (Arrington 2010, Ante 2011). The participating firms compete against each other in a "market," where customer choice is represented by the evaluation of event judges. These judges evaluate the output of each firm at the end of the competition, rewarding selected firms with prizes based on a number of pre-selected criteria. Though judging criteria may differ based on the nature of a particular hackathon, hackathons across contexts favor novelty in ideas and solution approaches. This makes hackathons a well-suited environment to study the Agile phenomenon; much like the features of software markets during Agile's inception, the hackathon environment prioritizes novelty and innovation.

Hackathon sponsors commonly provide mentors to participating firms throughout the competition. Given their non-evaluative, authority-free support role at hackathons, mentors present ideal facilitators of our iterative coordination treatment of stand-up meetings.

3.1.2 Competition Specifications In terms set by Google, competing firms developed a software application that provided a novel solution to some pro-social goal, e.g., a sustainability app to track personal carbon footprint or an app to collect data for NGO fieldworkers. Consistent with standard hackathon practice, firms chose the specific problem they wished to work on, provided it was in service of the general theme of the event. Prizes totaling $2,000 USD in monetary value were provided by Google to top-performing firms.[5]

In collaboration with Google, we recruited firms consisting of software engineers to compete in the hackathon. Competing firms were composed of upper-level undergraduate computer science majors from local universities and professional and freelance software developers. Individual participants qualified based on their prior collaborative software development experience, through a submitted portfolio of past projects. Participants registered together in firms of two to four members in a pre-event survey designed with our co-sponsor; the pre-event survey data also served as a source of control variables and for screening potential participants on the technical skills necessary to be productive during the hackathon.

Prior to the start of the competition, firms were randomly assigned into treatment and control groups. Table 1 displays results of t-tests for differences across treatment and control firms for each of these time-invariant firm characteristics drawn from the pre-event survey. We do not find any evidence of systematic bias in our randomization. In all, 38 firms competed in the hackathon, consisting of 112 participants (62 students and 50 professionals).

— Insert Table 1 About Here. —

While firms had flexibility to define the nature of their applications, they met two basic requirements for the competition. First, they were required to use a fixed development toolkit provided by Google. This finite software toolkit limits the product attributes firms can consider and potentially recombine to produce their applications (Levinthal 1997, Fleming 2001). By holding available technological inputs constant across treatment and control groups, we strengthen our ability to interpret the causal effect of our intervention. Second, to collect the detailed data over time on development processes, all firms were required to record their work over the course of the competition with the industry-leading open-source version-control software, Git. By tracking the emergence of software code, Git allows for the detailed measurement of software development activities over time.[6]

3.2 Experimental Procedure

Our experimental treatment was iterative coordination, implemented as regularly scheduled stand-up meetings. Our choice of treatment was directly inspired by the use of these regularly scheduled stand-up meetings in software development practice, often as part of broader "Agile" or "Scrum" methodologies.

3.2.1 Experimental Treatment Leveraging the natural features of the hackathon format, Google engineers who served as mentors to each firm facilitated stand-up meetings for iterative coordination. At the start of the treatment period, all firms, in both the treatment and control groups, were approached every two hours by their randomly assigned mentor, who was instructed to offer a null greeting in reference to an item on the schedule (e.g., "How was lunch?"). Each mentor appeared before an equal number of treatment and control firms. After this greeting, mentors visiting control firms conclude their interaction. In contrast, mentors visiting treatment firms would facilitate a short stand-up meeting asking treatment firms to answer three questions: (1) "What have you accomplished since your last check-in?"; (2) "What are your goals until the next check-in two hours from now?"; and (3) "What are your goals for the end of the day (and have they changed)?"[7] Per instruction, mentors did not provide any value judgments to firms during these stand-up meetings; rather, they simply served as facilitators for group discussion. To ensure the treatment closely reflected practice, we devised the three aforementioned questions after observing stand-up meetings used at Google; we further verified the external validity of these questions in interviews with other engineers from Google and peer firms that practice stand-up meetings.[8]

We built in a pre-treatment period of 2.5 hours in which no firms were treated. The inclusion of this pre-treatment period allows us to include firm fixed effects and run a generalized differences-in-differences regression model. As we shall address in our firm-minute analysis of the firm processes in Section 3.4, the firm fixed effects control for time-invariant quality differences between firms, bolstering causal identification in case there was any further unobserved time-invariant heterogeneity between firms not addressed by randomization of the treatment. After this pre-treatment period, the periodic stand-ups occurred every two hours until the close of the competition for treatment firms but not for the control firms. Treatment firms experienced three stand-up meetings over the course of the competition.

3.2.2 Other Experimental Design Considerations To maximize external validity and protect causal identification from the field experiment, we made several explicit efforts to limit the participant perception of mentor authority and to prevent participant awareness of heterogeneous treatment.

We minimized the perception of mentors as authority figures in three ways. First, it was clearly communicated to the participants that mentors absolutely did not serve as or communicate with the judges in the competition. Second, the mentors were demographically similar (e.g., age, professional background, etc.) to the participants, minimizing perceptions of authority enforced by differences in social status (Ashforth and Mael 1989, Lincoln and Miller 1979). Third, the mentors did not provide any unsolicited normative guidance to participants.

We ensured that participants remained unaware that treatment firms and control firms experienced different mentor interactions through a number of design decisions made in conjunction with our partner Google. The undesirable consequences of such awareness range from spillover effects from treatment to control (Duflo and Saez 2003), to Hawthorne effects where firms act differently due to their awareness of being observed for study (Levitt and List 2011). First, we physically separate the workspace of treatment and control to minimize the chance of across-condition interaction that might lead to awareness of different attention from mentors. Second, to reinforce perception of parity, the assigned mentors visit firms every two hours in both treatment and control, as previously noted. While the pure counterfactual control to our three-question iterative coordination treatment could conceivably involve no interaction with mentors every two hours in control, excluding control firms from any mentor visits would risk increasing awareness of differential mentor attention. In addition, keeping the mentor visit events uniform for both treatment and control firms has the desired effect of keeping constant the time for cycles of software development work. Discussions with Google made it evident that any mentor interaction could break up software development cycles in a way that may otherwise not occur. To isolate the causal effect of iterative coordination questions—without the potential confounding factor of different cyclicality—we ensured that both treatment and control would be visited by mentors on the same two-hour cycle. The Online Appendix provides further detail on the experimental design, including detailed floor plans of the physical space and mentor scripts.

We now discuss the data, statistical methods, and results of the field experiment looking at the effect of iterative coordination: first, a section on innovation outcomes, i.e., value and novelty; second, a section on potential processes, i.e., integration and specialization, which may lead to innovation.

3.3 Organizational Outcomes

We begin our analysis of iterative coordination on innovation by analyzing its effects on outcomes of *Value* and *Novelty*, two key dimensions of innovation (Amabile 1983, Kaplan and Vakili 2015).

3.3.1 Data Our dataset to study performance outcomes consists of a cross-section of firm project evaluation by expert judges after the end of the hackathon competition, combined with a set of covariates to serve as control variables collected through a pre-event survey.

Each firm was visited by a third-party panel of three judges to evaluate their projects at the end of the competition. These judges were not affiliated with the competition, and they were unfamiliar with our study design. Each judge had several years of work experience in the software industry and had both participation and judging experience in other hackathons prior to our event. The judges tested and interacted with the applications that the firms developed.

As part of the formal registration process for the hackathon, participants were asked to complete a short registration survey that was designed with our co-sponsor. Descriptions of variables measured for each firm in the survey, along with summary statistics and pairwise correlations, are provided in the previously discussed Table 1. To address across-firm heterogeneity across treatment and control firms not addressed by experimental randomization, we use the following variables as controls: educational experience (*Current Student*, *Graduate Degree*), various dimensions of software development experience (*GitHub*, *Software Development*, *Google Development*, *Prior Hackathons*), and firm size (*Firm Size*)—factors potentially correlated with end-of-competition firm outcomes.

3.3.2 Dependent Variables To measure outcomes of innovation, we use two measures capturing different dimensions of innovation for the applications developed by firms. Our first outcome measure is *Value*, which measures the extent to which a firm caters to its existing target customer base. Our second dependent variable, *Novelty*, captures new problem-solving approaches within the scope of Google's furnished app development toolkit. Judges scored each firm's final project along the two aforementioned outcome categories based on a Likert scale of 1 to 5, summarized with descriptions in Table 2. The use of these specific criteria to evaluate software applications had been validated by our co-sponsor, Google, from experience hosting prior hackathons. Furthermore, value and novelty capture independent components of innovation (Amabile 1983, Singh and Fleming 2010, Kaplan and Vakili 2015), as discussed eariler.

— Insert Table 2 About Here. —

3.3.3 Estimation Model To compare end-of-competition firm outcomes, we run cross-sectional OLS models with dependent variables for the evaluation categories regressed on an indicator variable for treatment, with firm control variables drawn from the pre-event survey listed in in Table 1. In addition, we include the dummy indicator *No Evaluation* to control for whether a firm officially submitted an application for judge evaluation, which commenced a half-hour after the competition officially closed. Regardless of participation in judge evaluation, all firms nonetheless stayed to the end of the competition, and their project code was observable to us throughout the competition.

3.3.4 Results Table 3 presents the effects of iterative coordination on final outcomes. Model 3-1 demonstrates that iteratively coordinating firms scored on average 0.614 points higher on *Value* than informally coordinating firms ($p < 0.01$). Removing firms that did not participate in judge evaluation from our sample, Model 3-3 shows that iteratively coordinating firms scored an average of 0.846 points higher on *Value* ($p < 0.01$). Supporting our findings of a positive association between iterative coordination and *Value* are Models 3-2 and 3-4, which include the full set of firm controls to control for observable heterogeneity not addressed by the experimental randomization.

In contrast, Model 3-5 indicates that iteratively coordinating firms scored approximately a half-point less than control firms on *Novelty* ($p < 0.10$), with a similar negative association in Model 3-7 ($p < 0.10$). Models 3-6 ($p < 0.05$) and 3-8 ($p < 0.05$) demonstrate the robustness of this result when including the full set of firm controls.

— Insert Table 3 About Here. —

3.3.5 Supplemental Analyses We conduct a number of additional tests to verify the robustness of these findings to alternative specifications and explanations. We confirm the direction and statistical significance of these main effects across these analyses and rule out alternative explanations. First, we find the same patterns in an ordered logit analysis mirroring Table 3. Second, we rule out the alternative story that there may be differences in productivity across teams by looking for possible differences in project completion by the end of the competition. Third, we do not find that the intervention has any effect on selection by firms into evaluation. Fourth, we confirm that the same pattern holds when we allow for firm characteristics as moderators. The Online Appendix presents the full results of these robustness checks in detail.

In measuring the effect of iterative coordination on innovation outcomes, we find the results are mixed: while iteratively coordinating teams develop products that are more valuable, these products are simultaneously less novel. This result is intriguing—despite the competitive context of the Google hackathon demanding novel solutions, our results demonstrate that firms treated by iterative coordination questions produce less-novel output.

3.4 Organizational Process

Given the results outlined in the previous section, we ask: What processes might iterative coordination be influencing that associates with more valuable yet less-novel output? In Section 2, we reviewed competing perspectives from the literatures on creativity and the behavioral theory of the firm, where the former literature predicts more-novel output and the latter predicts less-novel output from iterative coordination (March and Simon 1958, Cyert and March 1963). As our results are consistent with the prediction from the behavioral theory of the firm, we now examine whether the predicted mechanism of organizational specialization helps potentially explain our results.

We now dive into firm software code to unpack iterative coordination's influence on mechanisms of integration and specialization, both of which have long been studied by organizations scholars for their relationship to innovation (Lawrence and Lorsch 1967, Grant 1996).

3.4.1 Data: Software Development To study the effect of iterative coordination on innovation process, we analyze a balanced firm-minute panel with dependent variables measuring integration and specialization in the software development process based on our minute-by-minute tracking of the firms' updates of their software code through Git.[9] With each update timestamped to the minute, our novel empirical strategy achieves a precise level of granularity.

Our dependent variables measure specific actions in the software development process consistent with integration or specialization, as summarized in Table 4.[10]

— INSERT TABLE 4 ABOUT HERE. —

3.4.2 Dependent Variables To measure organizational knowledge integration, *Code Integration Action* consists of the stock count of actions taken by the firm to integrate software code into and within the firm's shared code base. This measure captures two types of convergent development efforts in facilitating an integrated codebase. First, individual software developers, who may specialize and write some code independently, must combine their individual code with the firm's shared code base to integrate it with the overall project. Second, developers may combine code that is already in the shared codebase, thereby further integrating aspects of the project. The required version-control software enables and tracks these integrative activities, which are a standard part of software workflow management.

As an additional measure for specialization, *Advanced API Specialization* measures a firm's use of non-required specialized and advanced application programing interface (API) procedures, protocols, and tools in their codebase. While teams were required to use a broader toolkit provided by our sponsor Google, there were several advanced API tools available to the developers for free that were encouraged but not required in the competition. These tools allow firms to use a number of advanced cloud-based features: analyze data using artificial intelligence capabilities, run virtual machines, conduct natural language processing, leverage remote graphics processing units (GPU) for machine learning and 3D visualization, and connect with internet-of-things (IOT) devices, among other functionality. We measure the use of these tools by identifying the number of "API calls" or "API requests" to these tools appearing in a firm's codebase. These tools require in-depth specialized knowledge to use, beyond the common knowledge developers would have coming into the competition. Moreover, these tools were only free in the context of our competition; they were available as paid enterprise software outside of the competition, making it likely that developers would not use them regularly prior to the competition. An optional tutorial on these advanced features was available to all hackathon participants.[11] Since *Advanced API Specialization* reflects advanced technical development beyond the expected standards, we use it to measure specialization.

3.4.3 Estimation Model We use these two dependent variables in a firm-minute panel to estimate the following differences-in-differences model:

$$Y_{it} = \beta(Treatment_i \times Post_t) + \alpha_i + \delta_t + \epsilon_{it}.$$

Y_{it} represents the dependent variables of *Code Integration Action* and *Advanced API Specialization*. $Treatment_i$ is an indicator variable taking a value of 1 for firms treated by iterative coordination, and $Post_t$ is an indicator variable equaling 1 after the completion of the first of three mentor check-ins. Our coefficient of interest is β, which estimates the effect of iterative coordination meetings on Y_{it}.[12] α_i is a firm fixed effect that controls for time-invariant unobserved confounding factors (e.g., the complexity of the firm's chosen problem, Google toolkit know-how, etc.), and δ_t is a minute fixed effect to control for potential shocks across all firms during the hackathon (e.g., the beginning of lunch service at the event). We cluster robust standard errors at the firm level.

3.4.4 Results Table 5 reports the results of regression analyses that test the effects of iterative coordination on integration and specialization. Model 5-1 reveals that treatment is positively and statistically significantly associated with *Code Integration Action*. That is, after stand-up meetings commence, iteratively coordinating firms conduct on average 2.351 more code integrations than control firms.

On the other hand, Model 5-2 displays a negative and statistically significant relationship between iterative coordination and *Advanced API Specialization* use. Specifically, iteratively coordinating firms conduct 1.124 fewer highly specialized uses of Google's advanced application development toolkit in the post-period.

— INSERT TABLE 5 ABOUT HERE. —

3.4.5 Supplemental Analyses We devise a number of additional tests to assess the robustness of these findings. First, we devise two additional measures of integration and specialization based on the underlying structure of the file hierarchies in the software code. These two measures are based on branching factors, a standard performance measure in the computer science literature (Knuth and Moore 1975, Baudet 1978, Muja and Lowe 2009). We find statistically significant results consistent with those reported in Table 5. We carry these measures through the other robustness checks.

Second, to ensure the robustness of our results relative to estimates of standard errors which may be underestimated due to serial correlation in long time-series panels, we follow Bertrand et al. (2004) and run our main analysis with observations collapsed to pre- and post-periods. We find statistically significant effects consistent with our findings in Table 5.

Third, given the cumulative nature of our treatment—three administered iterative coordination meetings—we assess iterative coordination's cumulative influence on integration and specialization after each stand-up meeting. A concern for the viability and interpretability of our results would arise if, for instance, iterative coordination's effects on integration and specialization were observed early in the post-period were not sustained through the rest of the total observation period. As we would expect, our estimates of the effect of iterative coordination over time identify larger (further from zero) point estimates with greater statistical significance in later periods.

Fourth, we consider the extent to which observed differences between iteratively coordinating firms and firms in control may be driven due to differences in underlying productivity. If, for instance, iterative coordination hindered productivity, this alternative mechanism may instead explain iterative coordination's negative relationship with organizational specialization. Nonetheless, we find that iterative coordination bears no significant effect, positive or negative, on a firm's raw productivity, mitigating this alternative explanation.

Fifth, we find no statistically significant difference in the amount of time that treatment and control firms spent in their mentor interactions and afterwards to regroup and get back to work. The time spent for these interventions is very short compared to the overall length of the competition.

Sixth, we assess the results relative to potential moderating firm characteristics. The full results of these six categories of additional analyses are provided in the Online Appendix.

3.5 Discussion of Primary Study

We now holistically consider how the empirical findings of the primary field experiment fit together. We first show that iterative coordination affects the innovative output of the firms we study, where it associates positively with novelty and negatively with value as assessed in the final products of each firm. We then turn to the granular software code data to understand what processes iterative coordination might impact. We find that iteratively coordinating firms take more development actions oriented towards integration, while investing less in advanced, novel uses of Google APIs, suggesting decreased specialization.

This collective set of findings provides empirical support for arguments consistent with behavioral theory of the firm. This literature would predict that the cost of such decreased specialization from iterative coordination would be decreased novelty in outcomes, as we found from the outcomes analysis in Section 3.3. To formally test an empirical connection between specialization and novelty, we perform a post hoc mediation analysis, reported in the Online Appendix.[13] This analysis suggests there is reason to believe that integration and specialization are potential mediators that link iterative coordination to value and novelty, respectively. Nonetheless, we recommend caution in broadly interpreting this particular post hoc finding: we do not exogenously vary mediators of *Code Integration Action* and *Advanced API Specialization*, which would represent the ideal empirical design for a causal mediation analysis.

While the results thus far explore iterative coordination's influence on innovation outcomes and some potential mechanisms through which iterative coordination might act on innovation, it leaves open a key question: what levers of control do managers have at their disposal to regulate iterative coordination practice and its outcomes? Understanding the levers available to managers provides the scholarly literature with an understanding of the boundary conditions of iterative coordination while also facilitating the application of our findings to managerial practice.

Drawing from our knowledge of heterogeneity in implementations of iterative coordination in practice, two primary levers of control come to mind. First, some organizations vary whether or not each iterative coordination meeting updates organizational goals by changing the questions posed during iterative coordination meetings. Second, organizations can vary the frequency with which these meetings take place, which affects the pace at which goals update. Together, these two levers—question composition and meeting frequency—vary the intensity of goal updating in an iteratively coordinating organization (Sutherland 2005). To unpack the influence of these two levers, we conduct a follow-on laboratory experiment that provides the statistical power needed to study these two levers, beyond what was possible in our field setting.

4 Follow-On Study: Product Development Laboratory Experiment

To complement our primary field experiment, we run a second experiment in the laboratory to study the influence of the frequency of iterative coordination meetings and their ability to define and update goals on innovation outcomes. In addition to helping unpack these two levers of managerial control for iterative coordination, a follow-on lab experiment yields a number of desirable features. First, it follows the best practice of prior work to combine field data with laboratory experiments (Stoop et al. 2012). While our primary field experiment provides the benefit of external validity and applicability, it represents a less-controlled experimental environment. The laboratory environment provides that control and precision. Second, we collect a broader set of data, not possible in the field, to build alternative measures that confirm our findings in the field and allow us to measure alternative mechanisms that may take place and either bound or rule them out. In particular, we address the extent to which the interventions, in the form of formal meetings, account for time that participants would otherwise be working, and whether and how the intervention affects the degree of coordination that would otherwise take place in between meetings.

4.1 Experimental Design

In the experiment, teams design a new dorm or apartment product concept for a manufacturer. The teams compete with other teams for a cash prize based on their proposed product. We implement this task based on prior work by Girotra et al. (2010).[14] We randomly assign teams into one of three experimental conditions described later, where we vary the implementation of the iterative coordination treatment.

4.1.1 Participant Sample We recruited 210 participants, drawn from the general population, to participate in this study in a behavioral research laboratory at a northeastern university. Sample size was determined a priori by running a power analysis, based on effect sizes from the primary study. We pre-registered this laboratory experiment online with the Open Science Framework.[15] We randomly assign participants into 70 teams of three individuals and randomly assign each of these teams to one of three conditions. Verifying the validity of this randomization, we find no statistically significant difference in individual characteristics across the three conditions.[16]

4.1.2 General Procedure After entering the laboratory and completing a pre-experiment survey, participants receive instructions on their product development task. We randomly assign participants into teams of three and separately escort each team to their own private room to begin the experiment. Each team worked in their own private room for sixty minutes. We provide teams with sketch paper to make preliminary individual drawings; each individual could write on the sketch paper with an unique-color pen assigned to the individual. Each room contains a board for the team to illustrate its final product submission.

After the 60-minute experiment, participants first complete a post-experiment survey. Participants then vote on which team's final product concept, among those in their session, is their favorite; they are barred from voting for their own product. Within a session, we assign all teams to the same experimental condition, so no team is unfairly advantaged for the prize. In addition to $25 USD in compensation to participate in the study, the team that receives the most votes from peers in the session wins a prize of $10 USD per person.

4.1.3 Conditions We design three experimental conditions, seeking to vary iterative coordination by the frequency of its meetings and the extent to which its discussion questions update organizational goals. With these two dimensions of variation in mind, we sought an experimental design that would allow us to simultaneously explore both dimensions, while minimizing the number of experimental conditions to maximize the number of teams per condition for statistical power; team-level experiments are especially expensive given that they require, in our case, three times the number of recruited participants for the equivalent power of an individual-level study.

To implement the experimental treatments, a member of the research team acts as a team mentor who visits each team intermittently to administer the stand-up meeting intervention(s).[17] The general structure and content of the mentor/participant interaction parallel what was used in the hackathon in the Primary Study, except for the question content and meeting frequency.

In Condition 1, the baseline condition, we subject teams to only one intervention, asking only Question 1 of an iterative coordination treatment: after 20 minutes from the start of the experiment, the mentor asks one question to each team, "What have you accomplished since the last check-in?", which updates all the members of the firm on progress towards a pre-existing shared goal. In Condition 2, teams only experience one intervention at 20 minutes into the experiment, as in Condition 1, but we vary the question composition to include the opportunity to define and update shared goals; the mentor also asks, "What have you accomplished since your last check-in?" and "What are your goals for the end of the day?" We do not ask them "What are your goals until the next check-in?" since there is no next check-in and it would be redundant with the question on the goals for the end of the day. In Condition 3, teams experience two interventions, one at 20 minutes and one at 40 minutes, and they address all iterative coordination questions mirroring those posed in the hackathon field experiment. Table 6 summarizes the questions asked in each mentor/team interaction.

— Insert Table 6 About Here. —

The comparison between Conditions 1 and 2 captures the lever of defining and updating shared goals to isolate the influence of questions pertaining to goal updates; Question 1 only entails describing work achieved in pursuit of a pre-existing goal. In contrast, whereas Conditions 2 and 3 hold constant the lever of defining and updating goals, what differs is the frequency of updating these goals: Condition 3 experiences an additional iterative coordination meeting at 40 minutes.

4.2 Data and Measures

Table 7 details the source, construction, and interpretation of our empirical measures, organized by the construct they intend to measure.

— Insert Table 7 About Here. —

We document the final product of each team, which was a product drawing on separate whiteboard. These final products were rated by two independent raters on the dimensions of *Value* and *Novelty*. Given high levels of inter-rater agreement, 0.86 and 0.80 respectively, an average of the two ratings for each dimension of product outcomes was used in analysis.

We collect and code the work output of each individual, over the course of the experiment, and each team, at the end of the competition. We separately track the preliminary design work done by each individual over the course of the experiment in the form of sketches on regular pieces of paper. We use these sketches to manually code a measure of individual specialization, *Individual Sketches*, which is the count of the pages of draft sketches generated by individuals.

We record the video and audio of each team throughout the course of the competition. As a measure of integration, *Time to Integrate* assesses how quickly teams begin the process of integrating their individual work into the final product. *Time to Integrate* simply reflects the time when a team first writes on the whiteboard where they are required to report their final project submission, in contrast to the individual sketch paper where they individually specialize on preliminary ideas. To build this measure, a research assistant watches the video and takes down the time stamp of the first moment when a dry erase marker touched the whiteboard; there is no ambiguity in the coding process. Since teams use the whiteboard space for the final product, the action of writing on the white board reflects integration of team knowledge into the final product. *Time to Integration* serves as a fairly salient and non-survey based indicator of integration that we can observe in the laboratory.

4.3 Results

In Table 8, we report the summary statistics and cross-sectional analysis of the results of this laboratory study that compares differences in the means of the measures between Conditions 1 and 2 and between Conditions 2 and 3.[18] The direction and statistical significance of these findings are preserved when we instead use an OLS regression model that contains indicators for Conditions 2 and 3, where the relationships of interest would be the coefficient on the Condition 2 indicator and the t-test of differences in the coefficients on the indicators for Conditions 2 and 3.

— Insert Table 8 About Here. —

4.3.1 Outcomes Our findings on the outcomes in terms of the characteristics of the final product are consistent with the findings in the primary field experiment. We demonstrate that both the ability to define and update organizational goals and the frequency of meetings have a statistically meaningful effect on the outcomes, consistent with two suggested levers of control for iterative coordination. With respect to *Value*, we find that including an opportunity to define and update shared goals (comparing Conditions 2 and 1) generates greater *Value* ($\beta = 0.473$, $p < 0.01$), a 0.786 standard deviation increase. The second lever of increasing frequency of meetings, as represented by Condition 3, generates greater *Value* than in Condition 2 ($\beta = 0.295$, $p < 0.05$), a 0.490 standard deviation increase. With respect to *Novelty*, we find the opposite result. The addition of questions to define and update goals (comparing Conditions 1 and 2) generates lower *Novelty* ($\beta = -0.342$, $p < 0.10$), a 0.494 standard deviation decrease. Increasing the frequency of meetings in Condition 3 generates a lower degree of *Novelty* than in Condition 2 with only one meeting ($\beta = -0.397$, $p < 0.05$), a 0.574 standard deviation increase.

4.3.2 Process We also find evidence on process consistent with our primary field experiment. With respect to the process of integration, measured by *Time to Integrate*, we find that including an opportunity to define and update goals in Condition 2 (versus Condition 1) and the higher frequency of meetings in Condition 3 (versus Condition 2) lead to faster *Time to Integrate* ($p < 0.05$ and $p < 0.05$ respectively), suggesting that the separate components of iterative coordination do lead to integration. The effect of the additional goal question (Condition 2 vs. Condition 1) amounts to 396 seconds (6.6 minutes) faster *Time to Integration*, which accounts for a 0.60 standard deviation reduction in how long it takes a firm to integrate ideas on the board, amounting to 11% of the total time (one hour) the team had available for the task. The effect of the additional intervention (Condition 3 vs. Condition 2) amounts to 376 seconds (6.3 minutes) faster *Time to Integration*, a 0.57 standard deviation reduction amounting to 10% of the total available time.

To evaluate specialization, we then use our data on the sketches generated by the individuals on each team as indicative of intermediate individual-level specialization activity. We find that both the additional goal update question in Condition 2 (versus Condition 1) and the additional meeting in Condition 3 (versus Condition 2) reduces the total count of *Individual Sketches* generated by the members of each team ($p < 0.05$ and $p < 0.01$, respectively). This result implies that the opportunity to define and update shared goals or increasing the frequency of meetings reduces the capacity of the team to be productive in preliminary individual specialization activity. The effect of the additional goal question (Condition 2 vs. Condition 1) amounts to 1.1 less sketches, a 0.69 standard deviation reduction in sketches or lost sketch productivity of about 0.61 person-hour. The effect of the additional intervention (Condition 3 vs. Condition 2) amounts to 1.2 less sketches, a 0.76 standard deviation reduction in sketches or lost sketch productivity of about 0.86 person-hour.[19]

4.3.3 Additional Findings We collect data for a number of supplemental analyses to shed additional light on the effects of iterative coordination. First, we find no evidence that iterative coordination affects the completeness of the final product. Second, we confirm—via a post-experiment survey—that iterative coordination positively associates with self-reported measures of coordination and negative associates with self-reported measures of specialization. Third, we use the video recordings to code both the duration of iterative coordination meetings and the time it takes teams to get back to work after a meeting. We find that the meetings and time afterwards take up a small amount of time (3.6% of total available time), and adding a second meeting (Condition 3 vs. Condition 2) takes up less than double the additional time of only one meeting. Fourth, we measure the oral communication that takes place in between iterative coordination meetings using the video recordings. We find that adding the additional question (Condition 2 vs. Condition 1) and adding the additional meeting (Condition 3 vs. Condition 2) lead to a statistically significant increase in the frequency of communication between meetings but not in the total amount of words being spoken. The Online Appendix reports the data collection methods and results for these analyses.

Together, these results suggest robustness to a number of alternate mechanisms. For instance, the time cost of (additional) meetings cannot account for the entire observed decrease in specialization among iteratively coordinating organizations. While an additional meeting mechanically reduces raw time for work, the incremental time cost of a meeting decreases, i.e., a subsequent additional meeting takes less time than a previous meeting. Similarly, while additional meetings increase raw latency for teams to resume their work, these latency costs diminish with increasing frequency of meetings.

4.4 Discussion of Follow-on Study

The empirical findings of the follow-on study replicate the main findings of the primary field study and shed light on two levers of managerial control for iterative coordination.

Consistent with the findings of the primary field study, the laboratory experiment shows that iterative coordination has a positive association with value and a negative association with novelty. Furthermore, iterative coordination associates with integrative activity—reflected in this study as the quicker integration of sketch material into the final product—while being negatively associated with individually specialized activity—the pages of individual draft sketches produced.

When testing two levers of managerial control of iterative coordination—namely the frequency of meetings and inclusion of opportunities to define and revisit long-term goals—we find that iterative coordination's observed effects on innovation amplify. That is, the addition of discussion questions enabling updates to organizational goals and the addition of meetings (increasing frequency) yields stronger positive effects on value and stronger negative effects on novelty. These two levers similarly amplify the positive effect on integration and negative effect on specialization associated with iterative coordination in general. These findings serve as additional evidence that goal iteration plays an essential role in how iterative coordination affects innovation. For business practice, these two levers allow mangers and organizations to tweak iterative coordination to generate the degree of value and novelty they desire.

5 Discussion and Conclusion

Through a field experiment and a laboratory experiment, we address the question: What effect does iterative coordination have on innovation? While the originators and practitioners of iterative coordination intend for the popular practice to drive innovation in organizations (Rigby et al. 2016b), the literatures on creativity (Amabile and Pratt 2016, Alexander and Van Knippenberg 2014) and the behavioral theory of the firm (March and Simon 1958, Knudsen and Srikanth 2014) offer competing perspectives on whether or not it actually results in more innovation. To study the relationship between iterative coordination and innovation, we embed a field experiment within a software development competition in partnership with Google. We find that while iteratively coordinating teams develop products that are more valuable, these products are simultaneously less novel. By tracking the underlying software code, we find that iteratively coordinating firms favor integration at the cost of specialization to create new knowledge. We then conduct a follow-on laboratory experiment to verify our finding from the field and to further unpack managerial levers to regulate iterative coordination. We find that including an opportunity to update organizational goals and increasing the frequency of meetings further amplifies the effects of iterative coordination.

We now detail three major contributions of this work: refining theoretical understanding of the relationship between goal coordination and innovation; introducing empirical methodology for the empirical study of organizations; and addressing recent calls from the literature to study new methods of organizing and their influence on innovation.

5.1 Goal Coordination and Innovation

Studying the managerially-relevant practice of iterative coordination sheds light on the theoretical understanding of the relationship between goal coordination and innovation in organizations. Our findings suggest that even in contexts prioritizing novelty in organizational goals, such as at hackathons and entrepreneurial settings more generally, increasing coordination and iterating on these goals paradoxically leads to less novel outcomes. This finding contrasts with the creativity literature, which predicts that frequently iterating on organizational goals is necessary to achieve novel outcomes (Alexander and Van Knippenberg 2014, Amabile and Pratt 2016, Cromwell et al. 2018). Instead, our results support the intuition of the behavioral theory of the firm, where increases in coordination of any form undermines specialization (Knudsen and Srikanth 2014). This conceptual perspective argues that specialization drives ultimately novel outcomes because each specialized member of the organization can push the boundaries of existing organizational knowledge (March and Simon 1958, March 1991, Taylor and Greve 2006).

Given the empirical findings, it is worth reflecting on why iterative coordination does *not* foster specialization of individual effort. It could be the case that frequently discussing organizational goals could allow a firm to specialize more effectively by avoiding situations where members duplicate each other's efforts (Ethiraj and Levinthal 2004) and allow members to divide and conquer when working towards new opportunities (Rivkin and Siggelkow 2003, Levinthal and Workiewicz 2018). Indeed, the promise of iterative coordination is an opportunity to take stock of existing opportunities and enable the firm to efficiently explore new spaces and opportunities (Sutherland 2005, Rigby et al. 2016a). However, instead of enabling its members to more efficiently experiment with new technologies (Burgelman 1994, Levinthal 2017), we find that iterative coordination's frequent goal updates privilege organizational efforts towards the integration of existing, rather than new, knowledge.

Furthermore, our findings have implications for organization designers who wish to promote innovation (Lee and Edmondson 2017). The practice of rapid iteration on organizational goals may in and of itself be insufficient to achieve the novelty necessary for innovation. Our findings contradict the intuition of prior work that privileges the importance of speed in innovation (e.g., Vesey 1991, MacCormack et al. 2001, Eisenhardt and Tabrizi 1995, Siggelkow and Rivkin 2005). While increasing the frequency of organizational goal updates might implicitly increase the speed of organization adaption (Sutherland 2005), this benefit does not necessarily translate to more innovation.

5.2 Methodological Contributions

We offer a number of methodological contributions to the organizations literature. First, we introduce hackathons as an externally valid research setting for organizations research. Hackathons bear a number of attractive features for researchers of strategy and organization, including their ability to represent the competitive dynamics of markets and the environment produced which values innovation and the generation of novel ideas and solutions. Furthermore, hackathons present structural features favorable for experimentalists, including the use of judges to represent the voice of customers and mentors who may serve as facilitators for experimental interventions.

In addition to introducing the use of hackathons, we are the first researchers in this literature, to the best of our knowledge, to track software development in real time using version-control software. The software setting offers a number of benefits in studying innovation over other settings: in particular, it provides the opportunity for researchers to observe integration and specialization over time. Software development tracking captures both the specialized work done by individuals and the integration of work across an organization. To understand the power of this approach, consider the process of designing a physical product such as a skateboard. An organization designing a skateboard may consider a number of competing designs for the skateboard's decks and wheels before committing to manufacture. To the researcher, many potential part integrations between the deck and wheels are unobserved. In contrast, the low prototyping cost in software development (e.g., changing lines of virtual code) (Thomke 2003) allows for a much broader scope of integrations. Other members of the organization and the researcher can observe these integrations. The organizations literature can leverage this trait of software development to increase the accuracy and precision of empirical studies of innovation.

5.3 New Methods of Organizing and Levers of Innovation

Finally, we respond to recent calls from the literature to study new phenomena in organizing—and whether these new phenomena offer new insights towards furthering innovation in organizations (Puranam et al. 2014, Burton et al. 2017). In describing the history of Agile methodology, one of its core founders, Jeff Sutherland, describes with co-authors the appeal of developing a management practice that blends "the benefits of both organizational separation and integration" (Rigby et al. 2016b). This astute observation relates to the classic tension between integration and specialization in the organizations literature: the two processes trade-off, but innovation nonetheless requires both processes (Lawrence and Lorsch 1967, Levinthal and Workiewicz 2018). Here, Sutherland and colleagues insinuate that the design of Agile helps blend the benefits of integration and specialization, raising a natural question of whether the methodology either effectively negotiates or even breaks this classic trade-off.

Our findings demonstrate that although Agile practitioners may have intended to create a methodology that helps negotiate the integration/specialization trade-off, the signature practice of iterative coordination decidedly favors a process of integration at the cost of specialization. More importantly, iterative coordination associates with decreased novelty, consistent with predictions from the behavioral theory of the firm. As a result, we argue that many organizations adopt iterative coordination for the wrong reasons: they adopt iterative coordination because they desire and expect more novelty, while ultimately receiving less novelty when putting iterative coordination into practice.

Nevertheless, there is reason for practitioners to be cautiously optimistic. Our follow-on laboratory experiment demonstrates that iterative coordination can be modulated to increase or decrease its effects on value and novelty. An organization can adjust how an organization iterates on its shared goals by changing the discussion questions or the meeting frequency. In this way, an organization can tailor the practice to their own needed levels of value and novelty by making the goal updating of iterative coordination more or less salient.

5.4 Limitations and Future Work

We conclude by noting limitations to the present study and opportunities for future work. First, we note that while the context in which iterative coordination is used deeply prioritizes novelty and innovation (Rigby et al. 2016b), iterative coordination questions themselves do not directly frame a need for novelty. Given our interest in evaluating iterative coordination as it is practiced, we do not include a treatment condition in the present study which frames one of the three iterative coordination questions with an additional requirement that the organizational goal (or its subsequent output) be novel. Varying the composition of iterative coordination questions to articulate an explicit need for novelty, value, or other outcomes is a promising area for future study.

In addition, across practical contexts, there may be heterogeneity in the effects of iterative coordination. For instance, the design of physical products and/or services, as opposed to software development, may have fundamentally different organizing needs, due to alternate environments of complexity and modularity inherent to the architecture of the offering (Ulrich 1995). If this is the case, we need to identify whether the generalizable properties of problems from these alternative contexts drive any possible differences. Future work could also examine in further depth how iterative coordination applies to larger organizations, as firm size is an important moderator for innovation (Damanpour and Aravind 2012).

Finally, the role of iterative coordination may implicitly shape how an organization selects among alternative innovations (Keum and See 2017). With each iterative coordination meeting, members have an opportunity to share results of their individual efforts to generate alternatives for the organization, after which the organization selects which alternatives to pursue or prioritize for further development. Thus, future inquiry should consider how the parameters of iterative coordination influence quality of innovation selection.

Endnotes

[1]Together with Ken Schwaber, Jeff Sutherland is credited with introducing Scrum and its daily practice of iterative coordination as a formal process at a 1995 research conference of the Association for Computing Machinery (ACM).

[2]In Agile practice, projects are time-bound within “sprints,” which break up software development cycles into smaller chunks, e.g., every two weeks. In Question 3, the use of the word “project” refers to the end of an Agile sprint.

[3]Kaplan and Vakili (2015) study the effect of recombination on innovation as measured by novelty and value. As a result of this decomposition of innovation, they find that novelty associates with local rather than distant recombination of alternatives. Similarly, Jung and Lee (2016) find that searching for original or novel recombinations is associated more with localness than with boundary spanning, refining prescriptions from the boundary-spanning literature.

[4]Google LLC is the largest subsidiary of Alphabet Inc.

[5]Academic research grants supported the operational expenses of the experiment, event, and venue.

[6]The Git, specifically GitHub, interface allows us to see which member contributed to which portion of the project over time. As each member writes code, they submit it to the shared GitHub repository that represents the body of code for the overall project by the firm.

[7]In the final check-in, two hours before the end of the competition, only the first and third questions were asked.

[8]Peer firms included Twitter, Inc. and Cisco Systems, Inc.

[9]Git is a free, open-source version-control system that enables the distributed software development. Version control keeps track of all the changes developers individually make to the firm’s “repository” of source code for a project. Git archives each firm’s source code repository at each update. Should errors be made during the development process, a developer can easily restore the firm’s repository to that of a prior “commit” or update, which stores a snapshot of the firm’s repository at the time the update was made.

[10]Online Appendix A3 presents the matrix of pairwise correlations.

[11]An additional requirement of Google LLC’s co-sponsorship of our event was the inclusion of a tutorial on advanced features of one of the competition’s required app development toolkits. This was offered to all firms late in the day, immediately after the second mentor check-in, and attendance was optional.

[12]$Treatment_i$ and $Post_t$ were not independently estimated in the model because they are collinear with the more precise fixed effects of α_i and δ_t, respectively.

[13]See Online Appendix A2.

[14]Online Appendix A4 presents the full details of the task.

[15]Available at: `https://osf.io/c7qmw/`

[16]To confirm the randomization, we confirm there are not statistically significant differences across the three experimental conditions in *Age*, *Gender*, *Graduate Education*, *Current Student*, *Any Experience*, and *Years of Experience*. Online Appendix A4 presents further detail on this randomization check.

[17]To avoid deception, the research team member introduces themselves as a member of the research team.

[18]Online Appendix A4 presents the correlation matrix.

[19]We calculate person-hour productivity in Condition 1 as the 1.8 pages of sketches generated by the average participant in an hour. In Condition 2, person-hour productivity is 1.4 pages of sketches per hour.

References

Alexander L, Van Knippenberg D (2014) Teams in pursuit of radical innovation: A goal orientation perspective. *Academy of Management Review* 39(4):423–438.

Amabile TM (1983) The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology* 45(2):357–376.

Amabile TM, Pratt MG (2016) The dynamic componential model of creativity and innovation in organizations: Making progress, making meaning. *Research in Organizational Behavior* 36:157–183.

Andreessen M (2011) Why software is eating the world. *Wall Street Journal* .

Ante SE (2011) Skype to acquire start-up GroupMe. *Wall Street Journal* .

Arrington M (2010) GroupMe, born at TechCrunch Disrupt, secures funding and launches. *TechCrunch* .

Ashforth BE, Mael F (1989) Social identity theory and the organization. *Academy of Management Review* 14(1):20–39.

Baudet GM (1978) On the branching factor of the alpha-beta pruning algorithm. *Artificial Intelligence* 10(2):173–199.

Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, et al. (2001) Manifesto for agile software development .

Bertrand M, Duflo E, Mullainathan S (2004) How much should we trust differences-in-differences estimates? *Quarterly Journal of Economics* 119(1):249–275.

Broussard M (2015) The secret lives of hackathon junkies. *The Atlantic* .

Brown SL, Eisenhardt KM (1997) The art of continuous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations. *Administrative Science Quarterly* 1–34.

Burgelman RA (1994) Fading memories: A process theory of strategic business exit in dynamic environments. *Administrative Science Quarterly* 39:24–24.

Burton RM, Håkonsson DD, Nickerson J, Puranam P, Workiewicz M, Zenger T (2017) GitHub: exploring the space between boss-less and hierarchical forms of organizing. *Journal of Organization Design* 6(1):10.

Cao L, Ramesh B (2007) Agile software development: Ad hoc practices or sound principles? *IT Professional* 9(2):41–47.

Chatterji AK, Findley M, Jensen NM, Meier S, Nielson D (2016) Field experiments in strategy research. *Strategic Management Journal* 37(1):116–132.

Critcher CR, Ferguson MJ (2016) "Whether I like it or not, it's important": Implicit importance of means predicts self-regulatory persistence and success. *Journal of Personality and Social Psychology* 110(6):818–839.

Cromwell JR, Amabile TM, Harvey JF (2018) An integrated model of dynamic problem solving within organizational constraints. *Individual Creativity in the Workplace*, 53–81 (Elsevier).

Cyert RM, March JG (1963) *A Behavioral Theory of the Firm* (Englewood Cliffs, NJ: Prentice Hall).

Damanpour F, Aravind D (2012) Organizational structure and innovation revisited: From organic to ambidextrous structure. Mumford M, ed., *Handbook of Organizational Creativity*, 483–513 (Oxford, UK: Elsevier).

Duflo E, Saez E (2003) The role of information and social interactions in retirement plan decisions: Evidence from a randomized experiment. *Quarterly Journal of Economics* 118(3):815–842.

Eisenhardt KM, Tabrizi BN (1995) Accelerating adaptive processes: Product innovation in the global computer industry. *Administrative Science Quarterly* 40(1):84–110.

Ethiraj SK, Levinthal DA (2004) Bounded rationality and the search for organizational architecture: An evolutionary perspective on the design of organizations and their evolvability. *Administrative Science Quarterly* 49(3):404–437.

Fleming L (2001) Recombinant uncertainty in technological search. *Management Science* 47(1):117–132.

Furr N, O'Keeffe K, Dyer JH (2016) Managing multiparty innovation. *Harvard Business Review* 94(11):76–83.

Gavetti G, Greve HR, Levinthal DA, Ocasio W (2012) The behavioral theory of the firm: Assessments and prospects. *Academy of Management Annals* 6(1):1–40.

Gilson LL, Shalley CE (2004) A little creativity goes a long way: An examination of teams' engagement in creative processes. *Journal of Management* 30(4):453–470.

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

Grant RM (1996) Toward a knowledge-based theory of the firm. *Strategic Management Journal* 17(S2):109–122.

Hoppmann J, Naegele F, Girod B (2019) Boards as a source of inertia: Examining the internal challenges and dynamics of boards of directors in times of environmental discontinuities. *Academy of Management Journal* 62(2):437–468.

Jung HJ, Lee JJ (2016) The quest for originality: A new typology of knowledge search and breakthrough inventions. *Academy of Management Journal* 59(5):1725–1753.

Kaplan S, Vakili K (2015) The double-edged sword of recombination in breakthrough innovation. *Strategic Management Journal* 1457(36):1435–1457.

Keum DD, See E (2017) The influence of hierarchy on idea generation and selection in the innovation process. *Organization Science* 28(4):653–669.

Knudsen T, Srikanth K (2014) Coordinated exploration: Organizing joint search by multiple specialists to overcome mutual confusion and joint myopia. *Administrative Science Quarterly* 59(3):409–441.

Knuth DE, Moore RW (1975) An analysis of alpha-beta pruning. *Artificial Intelligence* 6(4):293–326.

Langley MA (2017) Pulse of the profession. Technical report, Project Management Institute.

Lawrence PR, Lorsch JW (1967) Differentiation and integration in complex organizations. *Administrative Science Quarterly* 1(12):1–47.

Leckart S (2012) The hackathon is on: Pitching and programming the next killer app. *Wired* .

Leckart S (2015) The hackathon fast track, from campus to Silicon Valley. *New York Times* .

Lee MY, Edmondson AC (2017) Self-managing organizations: Exploring the limits of less-hierarchical organizing. *Research in Organizational Behavior* 37:35–58.

Levinthal DA (1997) Adaptation on rugged landscapes. *Management Science* 43(7):934–950.

Levinthal DA (2017) Mendel in the c-suite: Design and the evolution of strategies. *Strategy Science* 2(4):282–287.

Levinthal DA, Workiewicz M (2018) When two bosses are better than one: Nearly decomposable systems and organizational adaptation. *Organization Science* 29(2):207–224.

Levitt SD, List JA (2011) Was there really a Hawthorne effect at the Hawthorne plant? An analysis of the original illumination experiments. *American Economic Journal: Applied Economics* 3(1):224–38.

Lincoln JR, Miller J (1979) Work and friendship ties in organizations: A comparative analysis of relation networks. *Administrative Science Quarterly* 181–199.

Lu S, Bartol KM, Venkataramani V, Zheng X, Liu X (2019) Pitching novel ideas to the boss: The interactive effects of employees' idea enactment and influence tactics on creativity assessment and implementation. *Academy of Management Journal* 62(2):579–606.

MacCormack A, Verganti R, Iansiti M (2001) Developing products on "internet time": The anatomy of a flexible development process. *Management Science* 47(1):133–150.

Madjar N, Shalley CE (2008) Multiple tasks' and multiple goals' effect on creativity: Forced incubation or just a distraction? *Journal of Management* 34(4):786–805.

March JG (1991) Exploration and exploitation in organizational learning. *Organization Science* 2(1):71–87.

March JG, Simon HA (1958) *Organizations* (New York: Wiley).

Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP 2009—Proceedings of the 4th International Conference on Computer Vision Theory and Applications* 2(331-340):2.

Okhuysen G, Bechky BA (2009) Coordination in organizations: An integrative perspective. *Academy*

of Management Annals 3(1):463–502.
Okhuysen G, Eisenhardt KM (2002) Integrating in groups: How formal knowledge interventions enable flexibility. *Organization Science* 13(4):370–386.
Ott TE, Eisenhardt KM, Bingham CB (2017) Strategy formation in entrepreneurial settings: Past insights and future directions. *Strategic Entrepreneurship Journal* 11(3):306–325.
Pervin LA (1992) The rational mind and the problem of volition. *Psychological Science* 3(3):162–164.
Puranam P, Alexy O, Reitzig M (2014) What's "new" about new forms of organizing? *Academy of Management Review* 39(2):162–180.
Ries E (2011) The Lean Startup.
Rigby DK, Sutherland J, Takeuchi H (2016a) Embracing Agile. *Harvard Business Review* 94(5):40–50.
Rigby DK, Sutherland J, Takeuchi H (2016b) The secret history of Agile innovation. *Harvard Business Review* .
Rivkin J, Siggelkow N (2003) Balancing search and stability: Interdependencies among elements of organizational design. *Management Science* 49(3):290–311.
Schwaber K, Beedle M (2001) *Agile Software Development with Scrum* (Upper Saddle River, NJ: Prentice Hall).
Seijts GH, Latham GP, Tasa K, Latham BW (2004) Goal setting and goal orientation: An integration of two different yet related literatures. *Academy of Management Journal* 47(2):227–239.
Shalley CE (1991) Effects of productivity goals, creativity goals, and personal discretion on individual creativity. *Journal of Applied Psychology* 76(2):179.
Shalley CE (1995) Effects of coaction, expected evaluation, and goal setting on creativity and productivity. *Academy of Management Journal* 38(2):483–503.
Siggelkow N, Rivkin JW (2005) Speed and search: Designing organizations for turbulence and complexity. *Organization Science* 16(2):101–122.
Simon HA (1973) The structure of ill structured problems. *Artificial Intelligence* 4(3-4):181–201.
Singh J, Fleming L (2010) Lone inventors as sources of breakthroughs: Myth or reality? *Management Science* 56(1):41–56.
Stoop J, Noussair CN, Van Soest D (2012) From the lab to the field: Cooperation among fishermen. *Journal of Political Economy* 120(6):1027–1056.
Sutherland J (2004) Agile development: Lessons learned from the first scrum. *Cutter Agile Project Management Advisory Service: Executive Update* 5(20):1–4.
Sutherland J (2005) Future of scrum: Parallel pipelining of sprints in complex projects. *Proceedings of the Agile Development Conference* 90–102.
Taylor A, Greve HR (2006) Superman or the Fantastic Four? Knowledge combination and experience in innovative teams. *Academy of Management Journal* 49(4):723–740.
Thomke SH (2003) *Experimentation Matters: Unlocking the Potential of New Technologies for Innovation* (Harvard Business Press).
Ulrich K (1995) The role of product architecture in the manufacturing firm. *Research Policy* 24(3):419–440.
Van de Ven AH, Delbecq AL, Koenig R (1976) Determinants of coordination modes within organizations. *American Sociological Review* 41(2):322–338.
Vesey JT (1991) The new competitors: They think in terms of 'speed-to-market'. *Academy of Management Perspectives* 5(2):23–33.
Von Hippel E (2005) *Democratizing Innovation* (Cambridge, MA: MIT Press).
Zott C, Amit R, Massa L (2011) The business model: Recent developments and future research. *Journal of Management* 37(4):1019–1042.

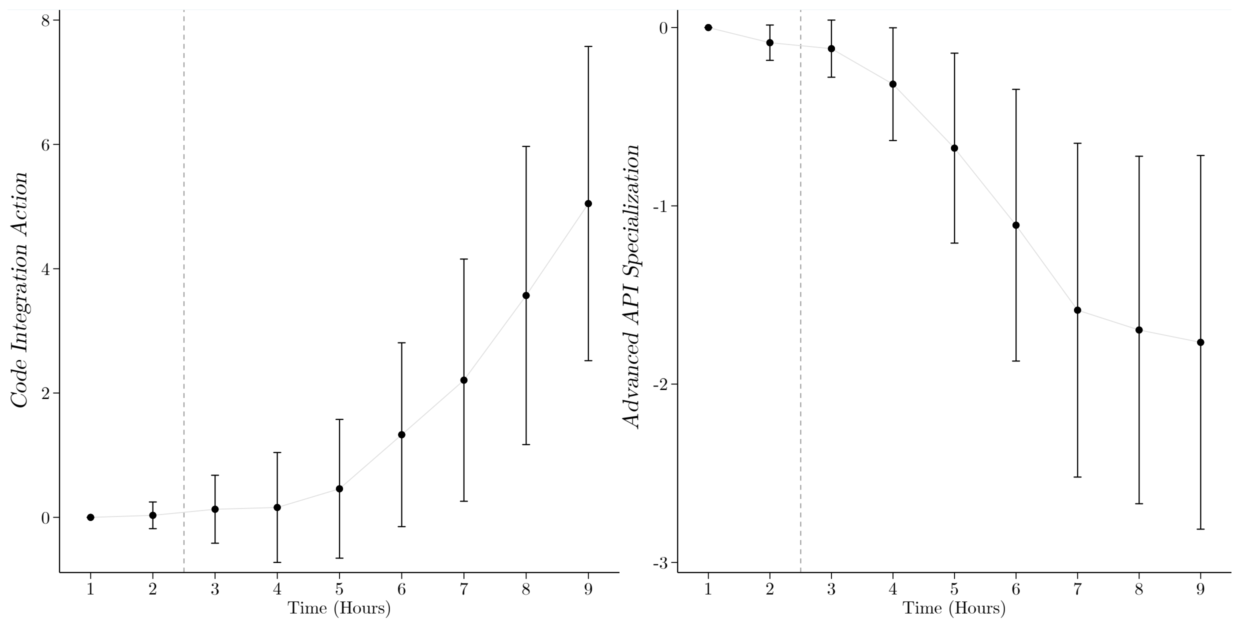


Figure 1: Primary Field Study: Effect on Process over Time. These four graphs depict the effect of the iterative coordination treatment over time. Each point estimate represents the difference in the level of the dependent variable between the treatment and control groups. The dotted grey line indicates the start of the treatment period. These graphs were constructed based on OLS regression estimates using an indicator variable for each hour of the experiment interacted with the treatment variable. The indicator variable for the first hour was necessarily omitted for estimation tractability, but shown here as the baseline term, set at a value of 0, i.e., equating the treatment and control groups. Firm and time (minute) fixed effects included. The 95% confidence intervals shown derive from robust estimates of standard errors clustered at the firm level.

Table 1: Primary Field Study: Firm Characteristics and Correlations. Means and standard deviations in parentheses for firm-level observations of the full sample ($N = 38$), treatment group, and control group. The Difference column shows a t-test of difference in means between the treatment group and control group, with standard errors in parentheses. The numbered columns to the right display pairwise correlations. *Current Student* is the firm mean of student status (with students taking a value of 1). *Graduate Degree* is the firm mean of educational experience (with Master's and Doctoral degrees taking a value of 1). *GitHub* is the firm mean of prior history using GitHub. *Google Development* is the firm mean experience with the development toolkits provided by Google. *Software Development* is the firm mean years of professional software development experience. *Prior Hackathons* is the firm mean of hackathons attended prior to the event. *Firm Size* is a count variable of the number of members in the firm.

	Sample				Pairwise Correlation						
Variable	Full	Treatment	Control	Difference	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1) Current Student	0.539	0.495	0.579	0.0838	1						
	(0.386)	(0.402)	(0.377)	(0.126)							
(2) Graduate Degree	0.360	0.375	0.346	-0.0292	-0.253	1					
	(0.344)	(0.334)	(0.361)	(0.113)							
(3) GitHub	0.901	0.954	0.854	-0.0995	0.227	-0.557	1				
	(0.192)	(0.138)	(0.223)	(0.0609)							
(4) Google Development	0.461	0.463	0.458	-0.00463	0.496	0.155	-0.049	1			
	(0.323)	(0.363)	(0.292)	(0.106)							
(5) Software Development	3.695	3.838	3.567	-0.271	-0.411	0.359	-0.124	-0.006	1		
	(3.839)	(3.661)	(4.083)	(1.264)							
(6) Prior Hackathons	1.825	1.662	1.971	0.309	0.104	-0.079	-0.033	0.238	0.349	1	
	(1.186)	(0.943)	(1.378)	(0.387)							
(7) Firm Size	2.947	2.722	3.150	0.428	0.132	-0.12	0.009	0.242	-0.361	-0.214	1
	(0.837)	(0.826)	(0.813)	(0.266)							

Table 2: Primary Field Study: Variable Definitions and Summary Statistics of Firm Outcomes from Judge Evaluation. Judges were asked to score each firm's final submission on a 1 to 5 Likert scale according to the criteria provided by our corporate co-sponsor Google.

Variable	Definition	Mean	SD
Value	How much does your project appeal to the intended market? (Likert scale 1 to 5)	2.553	1.796
Novelty	Does the project help solve the problem in a new and ambitious way? (Likert scale 1 to 5)	2.316	1.726

Table 3: Primary Field Study: Regression Analysis of Firm Outcomes from Judge Evaluation. Ordinary least squares (OLS) estimation of cross-sectional data at the firm level. Robust standard errors shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$. In estimates involving the full sample, *No Evaluation* takes a value of 1 for firms that decided to not undergo the judging process.

	Value				Novelty			
	(3-1)	(3-2)	(3-3)	(3-4)	(3-5)	(3-6)	(3-7)	(3-8)
Treatment Group	0.614**	0.546*	0.846**	0.661*	-0.499†	-0.692*	-0.687†	-0.960*
	(0.222)	(0.208)	(0.290)	(0.302)	(0.278)	(0.332)	(0.375)	(0.370)
Current Student		0.725*		0.991		0.145		0.638
		(0.353)		(0.650)		(0.390)		(0.793)
Graduate Degree		0.430		0.365		-0.458		-0.877
		(0.384)		(0.590)		(0.520)		(0.697)
GitHub		0.120		0.106		0.893		0.907
		(0.798)		(0.965)		(0.996)		(1.106)
Google Development		-0.207		-0.310		0.898†		1.006
		(0.410)		(0.813)		(0.498)		(0.695)
Software Development		-0.050†		-0.039		0.007		-0.003
		(0.028)		(0.041)		(0.040)		(0.048)
Prior Hackathons		-0.144		-0.182		-0.096		-0.046
		(0.087)		(0.107)		(0.124)		(0.162)
Firm Size		-0.113		-0.096		-0.222		-0.421†
		(0.102)		(0.149)		(0.202)		(0.223)
No Evaluation	-3.497***	-3.702***			-3.336***	-3.331***		
	(0.189)	(0.195)			(0.202)	(0.256)		
Constant	3.274***	3.590***	3.154***	3.456***	3.518***	3.284***	3.615***	3.782**
	(0.219)	(0.883)	(0.249)	(1.170)	(0.214)	(1.151)	(0.241)	(1.434)
R^2	0.874	0.922	0.261	0.574	0.774	0.824	0.117	0.454
Sample	Full Sample		Evaluation Only		Full Sample		Evaluation Only	
Observations	38	38	27	27	38	38	27	27

Table 4: Primary Field Study: Variable Definitions and Summary Statistics of Firm Process from Software Code. Dependent variables for firm process defined below with their conceptual interpretation. Observations are at the firm-minute level, with 20,520 firm-minute observations across 38 firms.

Variable (Interp.)	Definition	Mean	SD	Min	Max
Code Integration Action (Integration)	Count of actions taken by the firm to integrate software code into and within the firm's shared code base	1.954	3.316	0	20
Advanced API Specialization (Specialization)	Count of uses of non-required specialized and advanced application programing interface (API) procedures, protocols, and tools	0.700	1.474	0	7

Table 5: Primary Field Study: Regression Analysis of Firm Process from Software Code. Ordinary least squares (OLS) estimation of firm-minute level data. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	(5-1) Code Integration Action	(5-2) Advanced API Specialization
Treatment x Post	2.074*	-1.124**
	(0.878)	(0.408)
Firm FE	Yes	Yes
Time FE	Yes	Yes
R^2	0.456	0.335
Adjusted R^2	0.441	0.317
Firms	38	38
Observations	20,520	20,520
Level	Firm-Minute	Firm-Minute

Table 6: Follow-On Laboratory Study: Experimental Conditions. This table summarizes the experimental intervention design in each of the three conditions. Different sets of questions at different times were posed by the mentor in each intervention.

Time Elapsed	Condition 1 (Baseline)	Condition 2 (Question Composition)	Condition 3 (Number of Interventions)
20 Minutes	What have you accomplished since the last check-in?	What have you accomplished since the last check-in? What are your goals for the end of the day?	What have you accomplished since the last check-in? What are your goals until the next check-in? What are your goals for the end of the day?
40 Minutes			What have you accomplished since your last check-in? What are your goals for the end of the day? (And have they changed?)

Table 7: Follow-On Laboratory Study: Variable Definitions and Sources. Measures drawn from each team's *Final Output* design, *Video Recording* of their working session, and their *Individual Sketches*.

Variable	Definition	Source
Outcomes		
Value	Usefulness of the final product (Likert 1–5). Average of two independent rater assessments.	Final Output
Novelty	Novelty of the final product (Likert 1–5). Average of two independent rater assessments.	Final Output
Process		
Time to Integrate	Time in seconds into the experiment until the team began working on the final product on the board based on draft individual sketches.	Video Recording
Individual Sketches	Count of pages of draft sketches generated by individuals. Reflects total productivity of individual specialized work.	Individual Sketches

Table 8: Follow-On Laboratory Study: Summary Statistics and Cross-Sectional Analysis. The first three columns contain the mean and in parentheses the standard deviation of teams in each condition. The last two columns compare of Conditions 1 vs. 2 and Conditions 2 vs. 3, respectively, based on a t-test of the difference in means; the values reflect the difference in means and in parentheses the standard error, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	Sample				Difference in Means	
	Full	Condition 1	Condition 2	Condition 3	1 vs. 2	2 vs. 3
Outcomes						
Value	3.386	2.971	3.444	3.739	0.473**	0.295*
	(0.602)	(0.594)	(0.535)	(0.414)	(0.165)	(0.140)
Novelty	3.190	3.551	3.208	2.812	-0.342†	-0.397*
	(0.692)	(0.729)	(0.612)	(0.540)	(0.196)	(0.169)
Process						
Time to Integrate	1579.6	1969.3	1572.8	1196.9	-396.4*	-375.9*
	(659.5)	(496.6)	(771.8)	(427.4)	(190.2)	(183.1)
Individual Sketches	4.171	5.304	4.208	3.000	-1.096*	-1.208**
	(1.579)	(1.329)	(1.560)	(0.853)	(0.424)	(0.369)

A1 Online Appendix: Design of Primary Study (Software Field Experiment)

A1.1 Spatial Setup

Figures A.1 and A.2 depict the overall event space floor plan and detailed room floor plans, respectively. Participants in the control and treatment condition are in different rooms, so they never observe the mentor–participant interaction of participants in the other condition. There was limited opportunity for cross-condition social interaction during the competition; participants could interact prior to the competition, during registration and the welcome presentation, and after the competition during the dinner and the awards ceremony.

— Insert Figure A.1 About Here. —

— Insert Figure A.2 About Here. —

A1.2 Scripts for Mentor-Participant Interaction

Google mentors interacted with their assigned teams in treatment and control following this provided sample script shown in Figure A.3. The script distinguishes between which content was delivered to participants in both conditions versus only participants in the treatment condition.

— Insert Figure A.3 About Here. —

A2 Online Appendix: Outcomes from Primary Study (Software Field Experiment)

We document of a number of additional tests to verify the robustness of the main results presented on the firm outcomes.

A2.1 Nonlinear Estimation: Ordered Logit

For robustness, we conduct an additional analysis of firm outcomes utilizing an ordered logit model. Our measures of *Value* and *Novelty*, based on underlying Likert scores on a scale of 1 to 5, are ordinal rather than continuous. Thus, it may be appropriate to utilize a nonlinear estimation method such as ordered logit, which accounts for dependent variables such that the relative ordering of response values is known but the "distance" between them is not. For parsimony, we chose to report OLS estimates in the main paper in Table 3. To verify the robustness of those results, in Table A.1, we utilize the same dependent and independent variables used in the models in Table 3, except that we estimate the models using ordered logit regressions. We confirm the direction and statistical significance of the reported coefficients using this alternative model.

— Insert Table A.1 About Here. —

A2.2 Differences in Productivity: Project Completion

We now test whether our results may be explained due to differences in productivity caused by our experimental treatment. We measure productivity in terms of software code generated and completion of their overall software application. Differences in productivity may potentially be explained due to the perceptions of authority. We address this alternate mechanism below.

A potential perception among our hackathon participants of mentors as authority figures is neither likely nor theoretically necessary to enable our iterative coordination treatment. Advantages of traditional authority-based hierarchy include efficiency in coordinating tasks, yielding increased productivity (Lee and Edmondson 2017). Nonetheless, Lawrence and Lorsch (1967) argue that non-hierarchical coordination may be vested in a designated coordinator or integrator who has no formal authority over the individuals whose activities require coordination. Following this approach, we designed the mentor role to serve as facilitator rather than authoritative supervisor of the iterative coordination treatment.

If perceived authority were to be vested in mentors, we would anticipate differences in net productivity among treatment and control firms as the primary observable effect of perceived authority. However, with authority-less mentors, we would anticipate no observable differences in net productivity between treatment and control firms. We test for a possible effect on productivity in outcomes here. Later in Appendix A3, we test for an effect in the processes, where we evaluate productivity in the writing of software code.

We verify that our experimental treatment did not have a statistically significant effect on whether or not firms completed their software application project by the end of the experiment. In Table A.2, we utilize the same regression models as in Table 3 of the main paper, except that we use the dependent variable of *Completion*. The same judges assessed *Completion* on a five-point Likert scale in response to the question: "How far was the firm able to get towards completing and implementing the project?" There is no statistically significant relationship between *Completion* and our experimental treatment, and the standard errors bound the point estimates within zero.

— Insert Table A.2 About Here. —

A2.2.1 Visual Representation In Figure A.4, we visually present the coefficient estimates of *Customer Needs* and *Novelty* from Table 3 of the main paper and *Completion* from Table A.2 of the

Appendix; we display coefficient estimates from the odd-numbered models of these tables.

— Insert Figure A.4 About Here. —

A2.3 Selection into Evaluation

We now verify that our experimental treatment did not have a statistically significant effect on whether firms underwent evaluation, which accordingly suggests that any effect of our experimental treatment on whether or not a judge evaluated a firm did not drive our main finding on firm outcomes in Table 3. Consistent with the standard procedure of most hackathons, participants may choose to opt out of participating in the final assessment by the judging panel, and thereby remove themselves for consideration from the set of available awards. A potential explanation for the choice to opt out is the perception by the participating firm that they are unlikely to win any of the available awards, given the firm's *ex ante* private information about the quality of their project. To provide additional detail on the nature of the choice to opt out, we provide additional descriptive analysis in Table A.3 to document whether the choice to opt out may relate to our experimental treatment or observable characteristics of the participants. We find no statistically significant relationship between *Evaluation* and our experimental treatments, nor with any observable characteristics of the firms.

— Insert Table A.3 About Here. —

A2.4 Moderating Firm Characteristics

To identify boundary conditions and verify the robustness of our firm process analysis, we explore potential moderator variables. We focus on the potential moderators of *Firm Size* and *Graduate Degree*, part of the time-invariant firm characteristics used primarily as control variables in the firm outcomes analysis. These two variables fit particularly well with aspects of our theory and setting. We consider how they affect the role of iterative coordination in both firm outcomes here and firm process later in Appendix A3.

For *Firm Size*, prior work establishes that coordination costs increase with organization size and, in turn, that coordination costs limit integration activity by the organization (Van de Ven et al. 1976, Okhuysen and Bechky 2009). Thus, we would expect that firms with larger *Firm Size*, as compared to smaller, should benefit more from iterative coordination in terms of undertaking more integration, i.e., *System-Level Branching* and *Code Integration Action*. Based on our theory, this integration should translate to *Value*. On the other hand, a reduction in coordination costs may also translate to performance in novelty because specialized knowledge must be eventually integrated into the final product.

We consider *Graduate Degree* to represent the degree of specialized knowledge present in the firms: higher educational institutions select for those with knowledge and improve the knowledge of those who receive it. Our main process result suggests that iterative coordination is negatively associated with specialization. We expect that firms with greater *Graduate Degree* because they have more specialization "to lose," should engage in even less specialization, e.g., *Subsystem-Level Branching* and *Advanced API Specialization*.

We present results of this analysis in Table A.4. Given the limited sample size, statistical significance is difficult, but there are several key results to highlight. Verifying the robustness of our main findings, we preserve the significant positive and negative effects of the iterative coordination treatment on *Value* and *Novelty*, respectively, in the base terms (Columns 1–4), while we continue to not find a statistically significant effect on *Completion* and *Evaluation* (Columns 5–8). Second, in the full *Novelty* specification in Column 4, we find that *Treatment Group* × *Firm Size* has a positive effect on *Novelty* ($p < 0.05$), although this same estimate is not significant in the more parsimonious Column 3 specification but the estimate stays the same direction. Nevertheless, the interpretation of this result would be that firms with iterative coordination improve in novelty as their size increases, but the opposite is true for firms without. Naturally, an increase in firm size mechanically introduces specialist knowledge to the firm, but it also introduces coordination costs that can limit whether that specialist knowledge translates into novelty in the final output. Third, we find that *Firm Size* is negatively associated with *Completion* for firms without iterative coordination, but positively associated with those that do. This result further suggests that the role of alleviating coordination costs may be important and, in turn, iterative coordination could be more valuable for organizations with high coordination costs, i.e., larger ones. We do not find any statistically significant interactions for the dependent variable of *Value* (Columns 1–2), nor for the *Treatment Group* × *Graduate Degree*.

— Insert Table A.4 About Here. —

A3 Online Appendix: Processes from Primary Study (Software Field Experiment)

We document additional tests to verify the robustness of the main results presented on the firm process analysis. We utilize the same dependent variables as described in the main manuscript in

Table 4 along with two additional measures described below.

A3.1 Software Code Hierarchy

We now describe two additional measures of integration and specialization beyond to further confirm the robustness of the results reported in the main paper, which reported on the dependent variables of *Code Integration Action* and *Advanced API Specialization*, respectively. These two separate empirical measures derive from how file hierarchies in the software code reveal underlying knowledge creation in software development.

We take a knowledge-partitioning perspective, which facilitates the interpretation of knowledge integration and specialization in problems of coordinated exploration (Knudsen and Srikanth 2014). In software development, or product and strategy development more generally, specializing members must coordinate their search efforts and integrate their individual knowledge bases to identify high-performing architectures (Eisenhardt and Tabrizi 1995, Grant 1996). In problems of coordinated exploration, a member's knowledge is represented as a set of partitions of the overall knowledge state-space (Samuelson 2004). Individual members create knowledge such that "the more the partitions in a member's information structure, the greater his or her knowledge about the space" (Knudsen and Srikanth 2014, p. 417). Search, then, proceeds by "going through the current information partitions or if necessary by further partitioning the information structure." Taking this knowledge-partitioning perspective, we apply it to our context of software development.

In software development, developers instantiate knowledge partitions via the creation of partitions in the file hierarchy through directories and files. File hierarchies consist of two types of "nodes": directories, which contain a set of files, and files, which contain a set of code lines. Directories are combinations of constituent files, and each file represents a unique combination of lines of code, embodying a unit of knowledge. File hierarchies, such as the example in Figure A.5, map to design hierarchies as described in Baldwin and Clark (2000). Developers categorize lines of code into files and groups of files into directories to engage in the practice of "information hiding," where elements of a computer program that are most likely to change are purposefully segregated from the rest of the software program (Parnas 1972, Baldwin and Clark 2000). Thus, through information hiding, hierarchies segregate "visible information" from "hidden information," signaling "who has to know what" (Baldwin and Clark 2000, p. 75). In summary, the property of information hiding allows us to map the file hierarchy to abstract knowledge partitions.

— Insert Figure A.5 About Here. —

We distinguish between integration and specialization by documenting evidence for knowledge creation at the system level and the subsystem level of the file hierarchies, respectively. We argue that the level of knowledge creation at different levels of the file hierarchy signals the intended function of a module of code, exploiting a known benefit of modularity in general (Ulrich 1995).

To observe integration in the file hierarchy, we construct *System-Level Branching*, measured as the branching factor specifically for the top-level directory.[20] "System-level" files and directories are closest to the top-level directory, also known as the "root" directory. The average branching factor, a standard performance measure in the computer science literature (Knuth and Moore 1975, Baudet 1978, Muja and Lowe 2009), is the ratio of the number of files and directories below the focal directories at a given time. At the system level, there is a single focal directory, i.e., the top-level "root" directory.

High *System-Level Branching* reflects integration efforts. Files and directories at this level of the hierarchy serve an integrative role, clustering member attention and recombining elements of lower-level files. High-level files and directories are the most visible to all members across a software development firm. Members interface with and look at code starting from the root directory. Thus, for a member to draw attention from other members to her own code, she would place it closer to the top of the file hierarchy. In the broader knowledge state-space represented by the overall file hierarchy, system-level files and directories are a natural place to cluster system-wide attention in line with integration (Okhuysen and Eisenhardt 2002). Furthermore, system-level files and directories recombine elements of lower-level files (Baldwin and Clark 2000), performing a key function of integration (Henderson and Clark 1990, Albert 2018).

To observe specialization in the file hierarchy, we measure *Subsystem-Level Branching*, which calculates the average branching factor for sub-root directories across the file hierarchy, where specialized knowledge creation occurs. "Subsystem-level" files and subdirectories reside further down the file tree, below the root directory. *Subsystem-Level Branching* increases if firms increase the number of files per subsystem directory.[21]

Subsystem-level files and directories reflect an intention for information hiding consistent with specialization efforts. Knowledge created at a lower level "only affect[s] their own piece of the system, hence they can be changed without triggering any changes in distant parts of the system" (Baldwin and Clark 2000, p. 75). Thus, to avoid creating or exacerbating existing interdependencies, subsystem-level nodes form a natural place for specialization in knowledge creation. In the absence of coordination, members specialize by developing files at subsystem-level directories. For instance, in a perfectly uncoordinated organization of autonomously searching members (Gavetti et al. 2005), each member would own a personal subdirectory below the root directory within which they would conduct individual search. Figure A.5 illustrates a file hierarchy and the calculation of *System-Level Branching* and *Subsystem-Level Branching*, reflecting integrative versus specialist software development, respectively.

In the context of software development, knowledge integration at the system-level entails key parts of the architecture that touch all parts of the product. In contrast, specialization at the subsystem-level addresses a subsystem or smaller module or component on the product. An example of the relationship between system-level and subsystem-level is Microsoft Windows (system) and the Notepad application (subsystem), which has always been a core part of the overall Windows product. Thus, the terms here refer to the level at which the product is being altered.

Table A.5 provides the summary statistics for these two variables.

— Insert Table A.5 About Here. —

A3.1.1 Interpreting "File Hiding" When agents create files, which they hide, i.e., store, at lower levels of the file hierarchy, we interpret this as evidence of specialization. Lower-level files arise naturally when individuals specialize in the development of their own code, without coordination, regardless of whether the code is functional or ultimately used. In contrast, integration takes place at the higher level of the file hierarchy, where those higher system-level files serve as evidence of integration and require coordination among contributors of specialized lower-level files. This choice is a type of information hiding, a broader concept from computer science, whereby engineers hide lower-level specialized components to shift focus on integrating components at a higher level. At least in the context of software development, our key interpretational assumption is that the ultimate act of integration is an independent activity from the creation of specialized files. Thus, evidence of specialization is distinct from evidence of a lack of integration. More explicitly, we do not interpret lower-level file hiding as integration or not, and we do not interpret higher system-level files as specialization or not.

We now describe three cases in which lower-level (hidden) files can exist and constitute specialization. These three cases are comprehensively exhaustive in our setting. First, the vast majority of lower-level files serve a present active purpose, and they reflect specialization in the act of their hiding when they were created. The content of these files may then be integrated by higher-level files to be tied into the main project and used for a present active purpose. Second, some lower-level files contain an error that prevents the current active use. Independent of the error, these error-containing files still reflect the output of specialized activity, although due to time or resource constraints, were not revised to serve a present purpose. Third, some lower-level files may themselves be complete but not yet integrated into the full project. These non-integrated lower-level files may have been intended for integration that was not achieved because of time or resource constraints, or they may be slated for deletion but not yet deleted. These lower-level files were still created through specialization. Even for the files intended for deletion, the intention of deletion does not negate the fact that they were created in the first place.

The relationship between lower-level files in general and integration may be ambiguous. Although we do not use this data in our paper, the presence of the first type of lower-level file reflects potential integration, while the third type reflects a lack of integration.

A3.1.2 Results We apply the same empirical methodology as used to study *Code Integration Action* and *Advanced API Specialization*, reported in the main document. Table A.6 reports the findings. Model 1 shows that iterative coordination is positively and significantly associated with *System-Level Branching*, such that treatment firms maintained an average of 3.26 more integrative nodes in their file hierarchies in the post-period. To examine the relationship between iterative coordination and knowledge specialization, we turn to Model 2. We find that iterative coordination has a statistically significant, negative association with *Subsystem-Level Branching*, with treatment firms maintaining an average of 0.981 fewer nodes per subdirectory. This indicates decreased knowledge creation at the subsystem level or decreased specialization overall.

—— INSERT TABLE A.6 ABOUT HERE. ——

A3.2 Correlation Table

Table A.7 displays the correlation matrix of these variables. The relatively large correlations of these variables is due to the cumulative nature of the software development process and the long time-series of the underlying data (i.e., at the minute level).

—— INSERT TABLE A.7 ABOUT HERE. ——

A3.3 Standard Errors in Differences-in-Differences Analysis

We present an analysis to verify the robustness of the statistical significance of the main findings reported in Table 5 relative to potential inflation of statistical significance. We cluster the standard errors in our main analysis in Table 5 at the firm level. While minute-level estimates of our results provide more careful consideration of the dynamics of the search process in software development, the granularity of this data may incidentally inflate the statistical significance of our results. In particular, we may face a serial correlation challenge. Because our estimation relies on a long time-series, our dependent variable likely serially correlates positively. Moreover, as an intrinsic aspect of a differences-in-differences type model, our key independent variable, *Treatment* × *Post*, by definition changes very little within a firm over time (Bertrand et al. 2004). Therefore, in this supplemental analysis, we collapse the minute-level data to a single observation in the pre-period and a single observation in the post-period, taking the separate averages of the dependent variables for both periods. We confirm the statistical significance of our main findings at the firm-minute level, with identical coefficients and standard errors.

—— Insert Table A.8 About Here. ——

A3.4 Effect of Treatment over Time

We also explore whether there are time-varying effects of our treatment. Instead of a single *Treatment* × *Post* indicator variable, we construct three separate independent variables, each representing the interaction between *Treatment* and one of three two-hour-long periods after the initiation of the experimental treatment. Each of these three periods corresponds to the time windows in between the three stand-up meetings and the end of the experiment. We present this analysis in Table A.9.

We confirm the statistical significance ($p < 0.10$) and direction of coefficient estimates for the second and third periods of the experiment for all dependent variables, and we further find statistical significance in the first treatment period for *Subsystem-Level Branching*. These results imply that the observable effect of our treatment is driven largely by differences occurring late into the experiment, suggesting that firms must undergo treatment for a sufficient time or experience a sufficient number of stand-up meetings for observers to recognize a treatment effect.

—— Insert Table A.9 About Here. ——

A3.5 Differences in Firm Productivity: Net Software Generated

As anticipated, we find there are no statistically significant differences in the amount of software code written by firms due to our experimental treatment. We measure productivity in terms of the net software code written by the firms, measured as *Lines*. In Table A.10, we run an analysis resembling our main analysis of firm productivity in Table 5, except with dependent variables of *Lines* and $\ln(Lines + 1)$, a logged version of the *Lines* variable to account for skewness in the underlying measure. These results suggest that there are no observable differences in raw software writing productivity due to the experimental treatment.

— Insert Table A.10 About Here. —

We wish to exclude differences in overall productivity that may have occurred due to unobservable heterogeneity. To test the robustness of our main findings on firm process in Table 5, we now include the control variable $\ln(Lines + 1)$ to address unobservable time-variant heterogeneity across teams; firm fixed effects already control for unobservable time-invariant heterogeneity across teams. We confirm the statistical significance of our main findings. The statistically significant relationship between $\ln(Lines + 1)$ and the dependent variable is mechanically due to the cumulative nature of the software development process, where a firm in a later period with more *Lines*, relative to the same firm in the earlier period, would have more *System-Level* and *Subsystem-Level Branching*, take on more *Code Integration Action*, and utilize *Advanced API*.

— Insert Table A.11 About Here. —

A3.6 Meeting Duration and Post-Meeting Latency

For an organization with a fixed or limited amount of time available, the pure act of iterative coordination takes time that the organization could otherwise use for other purposes. In practice, the time that an organization dedicates to its formal coordination meetings comes directly out of the finite time resource of its human capital. An organization that conducts these meetings cannot automatically compel its employees to work longer to make up for the lost time in the experiment. Studying this category of practices requires that we allow iterative coordination to account for real time. Thus, integrating the intervention time aspect into the experiment would allow us to measure the effect of the experiment on organizational performance in the most realistic way possible.

To provide a sense of what magnitude of time is accounted for by iterative coordination, we provide an estimate of the time effect for each separate meeting and the total. We turn back to the raw software code data and event records. We take the difference between the meeting start times and the first time that each firm enters new software code immediately after the meeting. This difference, *Meeting Duration & Post-Meeting Latency*, represents the total effect of iterative coordination on the time available to each firm to work; each iterative coordination meeting takes up time in both the form of the duration of the actual meeting and the time it takes the firm to regroup and get back to work after the meeting is over.

In Table A.12, we find that there are no statistically significant differences in the *Meeting Duration & Post-Meeting Latency* across treatment and control firms. Nevertheless, the point estimates can provide some sense of magnitude to frame the possible effect. For meetings 1, 2, and 3, there were differences of 5.3, -4.6, and 0.8 minutes, respectively, between treatment and control. The difference in total time taken up by iterative coordination, as opposed to the counterfactual intervention, is about 1.5 minutes: out of the total competition time of nine hours (540 minutes), iterative coordination accounts for less than 0.3% of the total time available to each firm, which we interpret as being relatively small. Thus, in the context of this particular field experiment, any possible time effect from iterative coordination is likely not large or meaningful.

In the follow-on laboratory experiment, we document that the effect of the interruption itself, faced by control firms in this field experiment but not the Condition 1 teams in the laboratory, is also likely not large; see Table 8 of the main manuscript for more detail.

—— Insert Table A.12 About Here. ——

A3.7 Moderating Firm Characteristics

We consider the effect of these moderators on our analysis of firm process. Table A.13 presents the results of this moderation analysis, where we use *Firm Size* and *Graduate Degree* in an interaction term with the main independent variable. In Columns 7 and 9, we find that *Firm Size* positively moderates the effect of iterative coordination on *Code Integration Action* ($p < 0.01$). In Columns 11 and 12, we find that *Graduate Degree* negatively moderates the effect of iterative coordination on *Advanced API Specialization* ($p < 0.10$). We do not find statistically significant results on the moderator term for the dependent variables of *System-Level Branching* and *Subsystem-Level Branching*, although we do preserve the significance and sign of the iterative coordination treatment.

— Insert Table A.13 About Here. —

A3.8 Mediation Analysis

We conduct a mediation analysis to empirically measure integration and specialization as mediators between iterative coordination and the outcomes of value and novelty. We follow the practices for measuring mediation relationships as described in the strategy and macro-organizational literature, in contrast to methods applied by micro-organizational and social psychology scholars. Following the guidance of Shaver (2005), we run a generalized structural equation modeling (SEM) analysis using the function as defined in Stata. Our model structure and estimation assumptions are similar to recent work by Kaplan and Vakili (2015), who study the effect of patent characteristics on innovation outcomes.

We combine the firm-time panel data on firm process (capturing processes of integration and specialization) with the cross-sectional data on firm outcomes (capturing the outcomes of value and novelty in the final output of each team). We combine these into one cross-sectional dataset. We preserve the cross-sectional data as is. For the firm-time panel data, we take the measures of *Code Integration Action* and *Advanced API Specialization* at their final value in the last period of the experiment to capture the cumulative integration and specialization respectively undertook by each firm over the course of the experiment and reflected in their final output; this measurement optimizes on the relationship between the mediators and the actual output that is assessed by judges to create the outcome measures.

Our SEM model uses the iterative coordination *Treatment* as the main independent variable, *Code Integration Action* and *Advanced API Specialization* as mediator variables reflecting constructs of integration and specialization, respectively, and *Value* and *Novelty* as the dependent variables. We allow for both *Code Integration Action* and *Advanced API Specialization* to be mediators to both *Value* and *Novelty*. While the primary mediation relationship of interest would be of integration to value and specialization to novelty, our model set up allows for integration to be a mediator for novelty and specialization to be a mediator to value, in the interest of completely evaluating all possible relationships. Across all individual regressions in the model, we include the full set of firm control variables used in Table 4 of the main paper. We use robust standard errors.

We present the results of the mediation analysis in Table A.14, estimated as one SEM model. The first two columns capture the relationship between the mediators of *Code Integration Action* and

Advanced API Specialization with the iterative coordination *Treatment*. The last two columns show the effect of the mediators and *Treatment* on the main dependent variables of *Value* and *Novelty*, allowing for mediation in the structure of the first two columns. Consistent with the main paper, we find that *Treatment* has positive relationship with *Code Integration Action* and a negative relationship with *Advanced API Specialization*. Then allowing for this mediation, we find that *Code Integration Action* has a positive and statistically significant effect on *Value*, while *Advanced API Specialization* has a positive and statistically significant effect on *Novelty*. At the same time, *Treatment* no longer has a statistically significant effect on *Value* or *Novelty*, having now been mediated, although we preserve the direction of the signs as in the analysis from Table 4 of the main paper.

— Insert Table A.14 About Here. —

In aggregate, these results suggest that there is reason to believe that integration and specialization are mediators that link iterative coordination to value and novelty, respectively. That said, we recommend caution on interpreting these findings. We do not exogenously vary *Code Integration Action* and *Advanced API Specialization* in our field experiment, and using them as mediators necessarily introduces endogeneity into the analysis. In particular, the strength of the field experiment is our ability to exogenously vary the iterative coordination treatment, implying some degree of causality to our interpretation. We cannot say with the same certainty that these estimates represent a causal mediation relationship.

A4 Online Appendix: Follow-On Study (Product Development Laboratory Experiment)

A4.1 Detailed Experimental Procedure

We recruited 210 participants for a study at the behavioral research laboratory at a northeastern university. They participated in a ninety-minute session: fifteen minutes to obtain consent, complete a pre-experiment survey, issue instructions, and assign teams and locations; sixty minutes for the actual experiment; and fifteen minutes at the end to complete a post-experimental survey, present their products, and vote for a prize winner.

A4.1.1 Phase 1: Pre-Experiment (15 Minutes) Participants enter the lab and are asked to indicate their consent for the study. Then, participants complete a pre-experiment survey on their basic demographic characteristics and their relevant education and experience for the experimental task. After completing the survey, participants are orally informed that they will be randomly assigned to teams of three to complete a product development task; the recruitment materials and aforementioned consent form already inform participants that they will be assigned to teams of three. Participants then receive instructions on their team task, adapted from Girotra et al. (2010):

> *Imagine that you have been retained by a manufacturer of dorm and apartment products to identify new product concepts for the student market. The manufacturer is interested in any product that might be sold to students in a home-products retailer. The manufacturer is particularly interested in products likely to be appealing to students. These products might be solutions to unmet customer needs, or products that have great potential but do not yet exist in the market. These products may also offer better solutions than products that already exist in the market.*
>
> *Your task as a team is to develop one product to present to the manufacturer by the end of today's session. Each of you is provided material to individually sketch and develop your own ideas, which we highly encourage you to use. Note that the most successful products proposals involve input from all team members. We encourage you to consider many product ideas before settling on a final team product. You will be asked to present your final product to your team mentor at the end of today's session using the presentation slide. Only one presentation slide is provided to each team. In addition to sketching and explaining your idea on the presentation slide, be prepared to offer a three-sentence summary of your product idea to your team mentor. Your mentor will then use this three-sentence summary to pitch your product to the rest of the session.*

A4.1.2 Phase 2: Experiment (60 Minutes) After reviewing the task, participants in their randomly assigned teams situate themselves at an assigned team work station, i.e., a private conference room. Each participant receives paper on which to sketch their individual ideas, along with one whiteboard per team, on which they are required to sketch and describe their final product idea with a three-sentence summary.

Members of the research team act as team mentors who visit the teams intermittently to administer "stand-up" meetings. These team mentors introduce themselves as members of the research team to avoid deception, which is against laboratory policy.

The experiment takes place over several separate sessions, where multiple teams participate in each session. Each experiment session features one and only one of three experimental conditions. Teams within the same session all fall into the same experimental condition and receive the same intervention. The three experimental conditions and their procedures are described in the main manuscript.

At the end of the 60 minutes for the product development task, teams turn over all their individual sketches and their whiteboard with the final product. In addition, teams provide a three-sentence summary of their product design idea, which they are asked to record in bullet points on the whiteboard.

A4.1.3 Phase 3 Post-Experiment (15 Minutes)

Participants individually complete a post-experiment survey. After completing the survey, a member of the research team presents each team's product based on the whiteboard sketch and three-sentence product summary collected at the end of Phase 2. Teams in the session then vote on their favorite product; they are barred from voting for their own product. All teams within a session are assigned to the same experimental condition, and so no team is unfairly advantaged through this process. The winning team receives a prize worth up to $10 per person.

A4.2 Participant Characteristics and Randomization Check

In a pre-experiment survey, we measure several participant characteristics to confirm the validity of our experimental randomization. Table A.15 presents the summary statistics and an Analysis of Variance (ANOVA) test showing that there are no statistically significant differences in these characteristics.

— INSERT TABLE A.15 ABOUT HERE. —

A4.3 Additional Measures

Table A.16 summarizes additional measures collected in the follow-on laboratory study not reported in the main paper. Table A.17 reports the summary statistics and cross-sectional analysis of the measures defined in Table A.16.

— INSERT TABLE A.16 ABOUT HERE. —

— INSERT TABLE A.17 ABOUT HERE. —

A4.3.1 Completion

Based on the final project of each team, we code a measure of *Completeness*—following the same coding method for *Value* and *Novelty* described in the main document—with an inter-rater agreement of 0.89.

In Table A.17, we find no statistically significant difference in the *Completeness* of the final product, and the insignificant point estimate is relatively small.

A4.3.2 Coordination and Specialization Post-Experiment Survey

We survey each individual after the experiment. After the experiment, we collect individual retrospective interpretations of the organization of each team and the contributions of each team member. Table A.18 depicts the post-experiment survey questions posted to participants regarding their perception of coordination and specialization within their teams. We adopt a battery of widely used survey measures of coordination (*Effectiveness*, *Few Misunderstandings*, *Low Backtracking*, *Efficiency*, and *Low Confusion*) and specialization (*Group*, *Individual*, *Responsibility*, *Necessity*, and *Awareness*) as proposed by Lewis (2003).We follow exactly the set of questions proposed by Lewis (2003) and used widely in subsequent work. Participants responded on a Likert scale (1–5).

— Insert Table A.18 About Here. —

In Table A.17, we report the results of the self-reported survey-based measures of coordination (i.e., *Coor.:...*) and specialization (i.e., *Spec.:...*), we find that a consistent set of estimates suggesting that adding both the question in Condition 2 (versus Condition 1) and the additional meeting in Condition 3 (versus Condition 2) leads to greater self-reported coordination in the teams but lower individual specialization. The direction of the estimates remains consistent throughout these measures, i.e., all positive for coordination and negative for specialization as we go from Condition 1 to Condition 2 to Condition 3. We find statistical significance on several of these survey-based measures, but not on all of them.

A4.3.3 Intervention Duration We use the video recordings to measure the incidental implications of our experimental treatment on the available time of our subjects, as a function of the time spent in the meeting (*Meeting Duration*) and the time after the meeting it takes to get back to work (*Post-Meeting Latency*).

As is mechanically the case, adding an additional meeting leads to more total *Meeting Duration*. However, *Meeting Duration* in Condition 3 of 98.6 seconds (1.6 minutes) is less than double the 68.8 seconds (1.1 minutes) in Condition 2, suggesting there is some diminishing need for formal meeting time as additional meetings are added. The additional question in Condition 2 adds a relatively small amount of time, 29.8 more seconds than in Condition 1 ($p < 0.05$). While these two results suggest there is a time cost to engaging in formal meetings, we must first take this in light of the fact that these differences are relatively small as compared to the total time of the experiment (3600 seconds total), where even in the most saturated Condition 3, it only reflects 3.6% of the total available time ($(98.6 + 31.7)/3600$ seconds).

With respect to the *Post-Meeting Latency*, we do not find a statistically significant difference between Conditions 1 and 2. As expected, the additional meeting in Condition 3 (versus Condition

2) mechanically leads to additional *Post-Meeting Latency* of about 20.9 seconds ($p < 0.05$), but the two meetings of Condition 3 have less than double the *Post-Meeting Latency* time of 32.7 seconds in the one meeting of Condition 2, consistent with the diminishing need pattern we observe on *Meeting Duration*.

A4.3.4 Ad Hoc Communication To understand the effect of our interventions on the ad hoc communication that takes place, we take the video recordings and transcribe the oral communication of each team and identify individual speakers and the time stamps of all communication. We measure the frequency (*Ad Hoc Frequency*) and word count (*Ad Hoc Word Count*) of the oral communication that occurs outside of the interventions.

We find that both the addition of the question in Condition 2 (versus Condition 1) and the additional meeting in Condition 3 (versus Condition 2) leads to a statistically significant increase in the frequency of distinct exchanges, *Ad Hoc Frequency*, that take place outside of the formal meetings ($p < 0.1$ and $p < 0.1$, respectively). Each exchange is an individual's oral communication bounded by speech by other individuals on the team. The estimates on *Ad Hoc Word Count* show no statistically significant differences across the conditions, although the pattern of point estimates goes up as we add an additional question in Condition 2 and an additional meeting in Condition 3.

We find no evidence that iterative coordination reduces the ad hoc communication that takes place outside of the formal meetings. Instead, the result on *Ad Hoc Frequency* suggests that iterative coordination increases the need (or at least realization of the need) to increase communication for interdependent integration purposes, where more meetings essentially reflect greater back-and-forth conversation between the members of the team.

While iterative coordination meetings may theoretically substitute for ad hoc meetings that would otherwise occur in between iterative coordination meetings, our results demonstrate the opposite. That is, instead of finding decreased frequency of ad hoc meetings and decreased volume of ad hoc words exchanged, we find that the frequency of meetings increases for the same (null effect) volume of words exchanged (for Conditions 2 and 3 relative to that of Condition 1). These results suggest that by updating the goals of teams via more frequent, goal-oriented iterative coordination meetings, the team may democratize its member contributions by involving greater exchanges across the team, for the same volume of words exchanged. In this case, increasing coordination in flat forms may help preserve theorized benefits of autonomy and democratization of contributions (Lee and Edmondson 2017).

A4.4 Correlation Table

Table A.19 presents the matrix of pairwise correlations for all measures in the follow-on laboratory study.

—— Insert Table A.19 About Here. ——

Appendix References

Albert D (2018) Organizational module design and architectural inertia: Evidence from structural recombination of business decisions. *Organization Science* 29(5):890–911.

Baldwin C, Clark KB (2000) *Design Rules: The Power of Modularity* (Cambridge, MA: MIT Press).

Baudet GM (1978) On the branching factor of the alpha-beta pruning algorithm. *Artificial Intelligence* 10(2):173–199.

Bertrand M, Duflo E, Mullainathan S (2004) How much should we trust differences-in-differences estimates? *Quarterly Journal of Economics* 119(1):249–275.

Eisenhardt KM, Tabrizi BN (1995) Accelerating adaptive processes: Product innovation in the global computer industry. *Administrative Science Quarterly* 40(1):84–110.

Gavetti G, Levinthal DA, Rivkin JW (2005) Strategy making in novel and complex worlds: The power of analogy. *Strategic Management Journal* 26(8):691–712.

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

Grant RM (1996) Toward a knowledge-based theory of the firm. *Strategic Management Journal* 17(S2):109–122.

Henderson RM, Clark KB (1990) Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly* 35(1):9–30.

Kaplan S, Vakili K (2015) The double-edged sword of recombination in breakthrough innovation. *Strategic Management Journal* 1457(36):1435–1457.

Knudsen T, Srikanth K (2014) Coordinated exploration: Organizing joint search by multiple specialists to overcome mutual confusion and joint myopia. *Administrative Science Quarterly* 59(3):409–441.

Knuth DE, Moore RW (1975) An analysis of alpha-beta pruning. *Artificial Intelligence* 6(4):293–326.

Lawrence PR, Lorsch JW (1967) Differentiation and integration in complex organizations. *Administrative Science Quarterly* 1(12):1–47.

Lee MY, Edmondson AC (2017) Self-managing organizations: Exploring the limits of less-hierarchical organizing. *Research in Organizational Behavior* 37:35–58.

Lewis K (2003) Measuring transactive memory systems in the field: Scale development and validation. *Journal of Applied Psychology* 88(4):587–604.

Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP 2009—Proceedings of the 4th International Conference on Computer Vision Theory and Applications* 2(331-340):2.

Okhuysen G, Bechky BA (2009) Coordination in organizations: An integrative perspective. *Academy of Management Annals* 3(1):463–502.

Okhuysen G, Eisenhardt KM (2002) Integrating in groups: How formal knowledge interventions enable flexibility. *Organization Science* 13(4):370–386.

Parnas DL (1972) On the criteria to be used in decomposing systems into modules. *Communications of the ACM* 15(12):1053–1058.

Samuelson L (2004) Modeling knowledge in economic analysis. *Journal of Economic Literature* 42(2):367–403.

Shaver JM (2005) Testing for mediating variables in management research: Concerns, implications, and alternative strategies. *Journal of Management* 31(3):330–353.

Ulrich K (1995) The role of product architecture in the manufacturing firm. *Research Policy* 24(3):419–440.

Van de Ven AH, Delbecq AL, Koenig R (1976) Determinants of coordination modes within organizations. *American Sociological Review* 41(2):322–338.

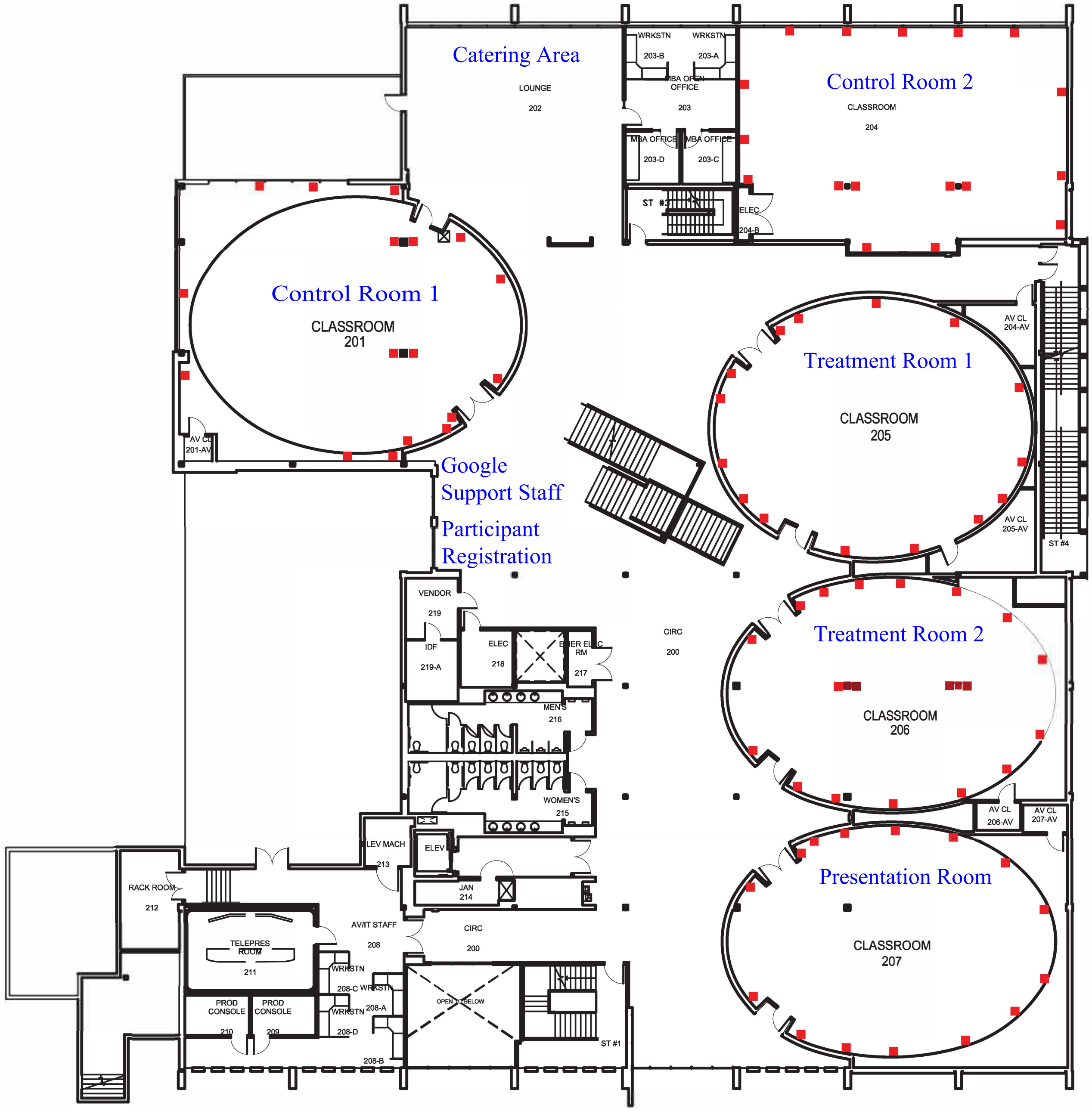
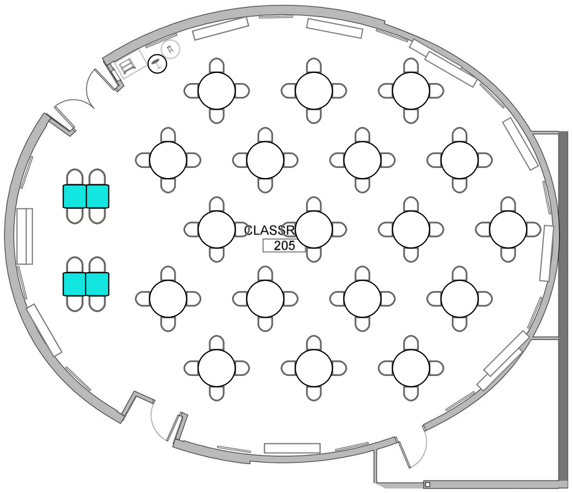
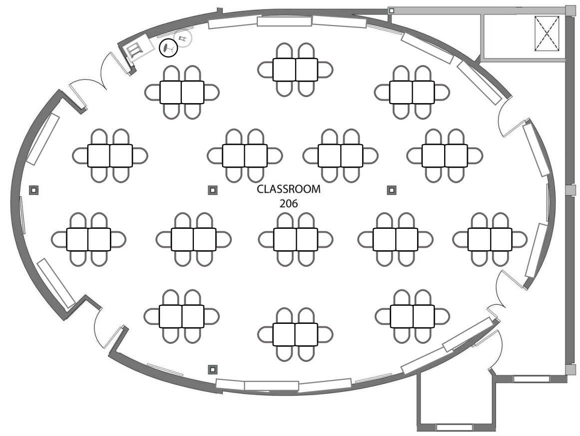


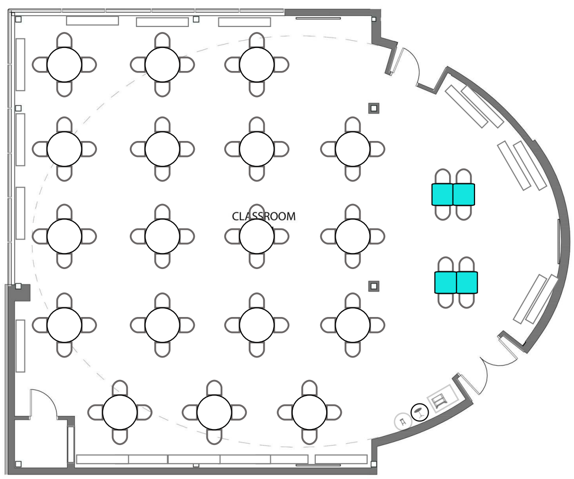
Figure A.1: Primary Field Study: Overall Floor Plan. This diagram depicts the overall floor plan of the space where the "hackathon" field experiment took place. Participants registered at tables located at *Participant Registration*. The treatment and control condition where divided into separate rooms, with two rooms for each condition. Lunch and dinner were served in the *Catering Area*; for lunch there was no public seating available, and participants brought their lunches back to eat at the tables where they worked. Staff stored their personal items and rested at the *Google Support Staff* station.



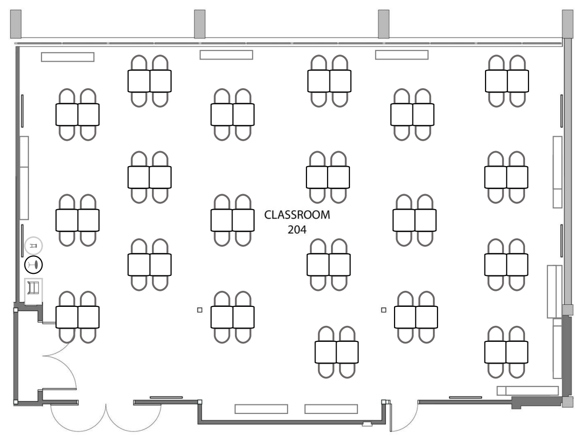
(a) Treatment Room 1



(b) Treatment Room 2



(c) Control Room 1



(d) Control Room 2

Figure A.2: Primary Field Study: Floor Plan for Treatment and Control Rooms. These detailed floor plans show the layout of tables where participants work in teams. There are more tables than teams. Participants in the control and treatment condition are assigned to different rooms, so they never observe the mentor/participant interaction of participants in the other condition.

Mentor Introduction (Prior to Competition)
Hello, my name is [NAME] and I'll be your mentor for today! I hope you're excited to participate in today's hackathon. First, let's run through some logistics. To participate in today's competition, we are asking each team to use Github version-control. Has your team successfully set up on Git? [Pause and verify.] Great! We will meet at your assigned table every two hours for a team check-in. Please be sure to be at your table at these times.

[Treatment only.] *At each of these meetings, I will be asking you to consider three check-in questions as a team. These questions are solely meant to help your team's process:*

- *What have you accomplished since your last check-in?*
- *What are your goals until the next check-in?*
- *What are your goals for the end of the day (and have they changed)?*

As mentioned in the welcome presentation, as mentors, we are not involved in the judging process. We are simply here to help with whatever you may need. Let us know if you have any questions! If you can't find me, feel free to ask any Googler for help. Good luck, and see you again at the first meeting!

Meeting 1
Hello there! How's it going? I'm here for the first check-in. Are you enjoying the hackathon so far? [Pause for response.] Excellent!

[Treatment only.] *Let's go through the questions I mentioned earlier:*

- *What have you accomplished since the beginning of today's competition?*
- *What are your goals until the next check-in?*
- *What are your goals for the end of the day (and have they changed)?*

Let a Googler know if you have any questions. See you in two hours!

Meeting 2
Hello there! How's it going? I'm here for the second check-in. How was lunch? [Pause for response.] Good!

[Treatment only.] *Let's go through the check-in questions I mentioned earlier:*

- *What have you accomplished since your last check-in?*
- *What are your goals until the next check-in?*
- *What are your goals for the end of the day (and have they changed)?*

Let a Googler know if you have any questions. See you in two hours!

Meeting 3. Hello there! How's it going? I'm here for the third and final check-in. Are you excited for the end of the hackathon? [Pause for response.]

[Treatment only.] *Let's go through the check-in questions I mentioned earlier:*

- *What have you accomplished since your last check-in?*
- *What are your goals for the end of the day?*

Let a Googler know if you have any last-minute questions. Good luck!

Figure A.3: Primary Field Study: Google Mentor Scripts. Google mentors interacted with their assigned teams in treatment and control following this provided sample script. The mentors only spoke the portions of script in *italics* when interacting with teams in the treatment condition and not with those in the control condition. Other instructions for the mentor are shown in brackets.

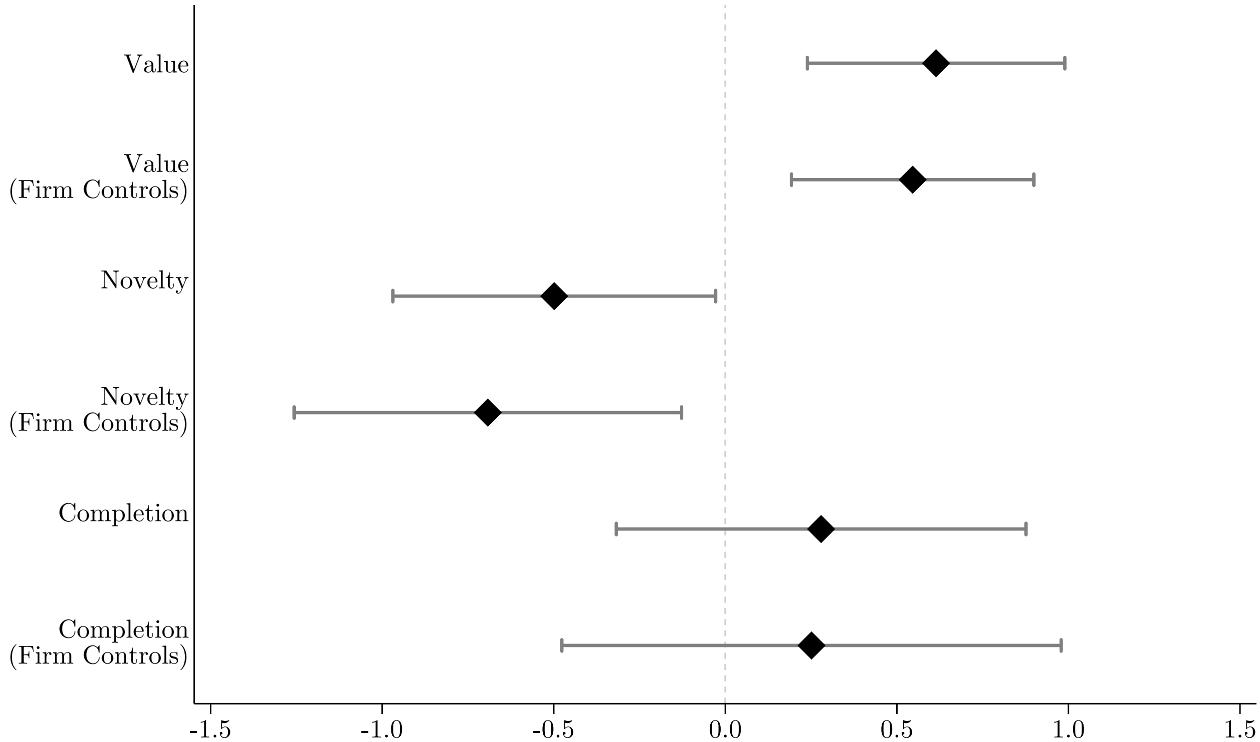


Figure A.4: Primary Field Study: Firm Outcomes. This figures plots the point estimates and 90% confidence internals for the analysis of the effect of the experimental treatment on firm outcomes. All estimates include the full set of observations, and the estimates labeled as "Firm Controls" include the full set of firm-level control variables.

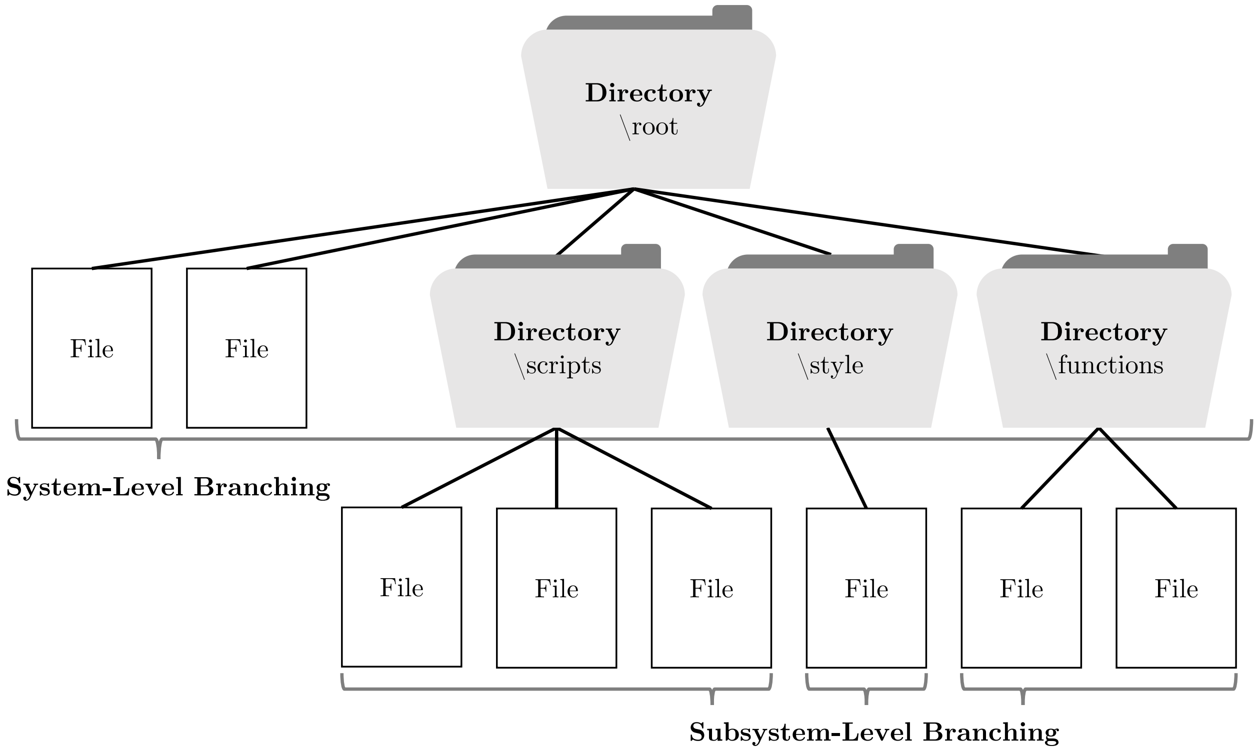


Figure A.5: System-Level versus Subsystem-Level Branching. To illustrate the measurement of *System-Level Branching* and *Subsystem-Level Branching*, we present an illustrative file hierarchy. *System-Level Branching* takes a value of $\frac{(2+3)}{1} = 5$, because there are 2 files and 3 directories at the system level, i.e., below the "root" directory, divided by 1 root directory. *Subsystem-Level Branching* takes a value of $\frac{3+1+2}{3} = 2$, because there are $3 + 1 + 2 = 6$ files at the subsystem levels contained within 3 directories.

Table A.1: Primary Field Study: Regression Analysis of Firm Outcomes Using Ordered Logit. This analysis resembles the main firm outcome analysis of the paper but instead estimated using an ordered logit model, which has favorable properties for Likert-scale discrete dependent variables. Robust standard errors shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	Customer Needs				Novelty			
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Treatment Group	2.310*	2.310*	2.537*	2.537*	-1.383†	-1.383†	-2.989*	-2.989*
	(0.946)	(0.951)	(1.091)	(1.097)	(0.802)	(0.806)	(1.339)	(1.346)
Current Student			4.305†	4.305†			2.370	2.370
			(2.456)	(2.470)			(2.022)	(2.033)
Graduate Degree			1.968	1.968			-2.662	-2.662
			(2.117)	(2.129)			(1.894)	(1.904)
GitHub			1.191	1.191			2.477	2.477
			(4.100)	(4.123)			(2.258)	(2.271)
Google Development			-1.325	-1.325			2.363*	2.363*
			(3.037)	(3.054)			(1.427)	(1.435)
Software Development			-0.159	-0.159			-0.011	-0.011
			(0.149)	(0.150)			(0.102)	(0.103)
Prior Hackathons			-0.738†	-0.738†			-0.019	-0.019
			(0.404)	(0.406)			(0.428)	(0.430)
Team Size			-0.318	-0.319			-1.183*	-1.183*
			(0.616)	(0.619)			(0.563)	(0.566)
No Evaluation	-40.334***		-43.187***		-38.875***		-44.855***	
	(0.860)		(2.513)		(0.684)		(3.050)	
Log Likelihood	-28.29	-28.29	-20.63	-20.63	-35.07	-35.07	-28.06	-28.06
Pseudo R^2	0.488	0.127	0.627	0.364	0.413	0.0484	0.530	0.239
Estimation	Ord. Logit	Ord. Logit	Ord. Logit	Ord. Logit	Ord. Logit	Ord. Logit	Ord. Logit	Ord. Logit
Sample	Full	Evaluation	Full	Full	Full	Evaluation	Full	Full
Observations	38	27	38	27	38	27	38	27

Table A.2: Primary Field Study: Regression Analysis of Completion. This analysis assesses the effect of the experimental treatment on the degree of *Completion* of the final submitted software application. Judges assessed *Completion* on a five-point Likert scale in response to the question: "How far was the firm able to get towards completing and implementing the project?" *Completion* has a mean of 2.000 and a standard deviation of 1.660. Ordinary least squares (OLS) estimation of cross-sectional data at the firm level. Robust standard errors shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	Completion			
	(1)	(2)	(3)	(4)
Treatment Group	0.279	0.385	0.251	0.336
	(0.353)	(0.484)	(0.428)	(0.594)
Current Student			0.336	0.917
			(0.575)	(0.932)
Graduate Degree			-1.065	-1.640†
			(0.649)	(0.876)
Github			-0.223	0.013
			(1.206)	(1.393)
Google Development			-0.127	-0.352
			(0.617)	(0.922)
Software Development			-0.042	-0.043
			(0.058)	(0.076)
Prior Hackathons			-0.094	0.021
			(0.147)	(0.290)
Team Size			-0.169	-0.195
			(0.208)	(0.300)
No Evaluation	-2.772***		-2.929***	
	(0.257)		(0.306)	
Constant	2.670***	2.615***	4.012***	3.640
	(0.319)	(0.367)	(1.426)	(2.193)
R^2	0.614	0.0249	0.699	0.362
Estimation	OLS	OLS	OLS	OLS
Sample	Full	Evaluation	Full	Evaluation
Observations	38	27	38	27

Table A.3: Primary Field Study: Regression Analysis of Selection into Evaluation. This analysis assesses the effect of the experimental treatment on whether firms select into evaluation by judges at the end of the competition. The dependent variable *Evaluation* is an indicator variable taking a value of 1 if the firm underwent judging, and 0 otherwise. The first two models are estimated using ordinary least squares (OLS) regression, and the last two models are estimated using a logit regression. Robust standard errors shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	Evaluation			
	(1)	(2)	(3)	(4)
Treatment Group	0.128	0.107	0.539	0.539
	(0.149)	(0.183)	(0.846)	(0.846)
Current Student		-0.293	-1.590	-1.590
		(0.313)	(1.654)	(1.654)
Graduate Degree		-0.050	-0.366	-0.366
		(0.303)	(1.466)	(1.466)
Github		0.111	0.538	0.538
		(0.401)	(1.986)	(1.986)
Google Development		0.288	1.611	1.611
		(0.380)	(1.912)	(1.912)
Software Development		0.003	0.017	0.017
		(0.028)	(0.159)	(0.159)
Prior Hackathons		0.048	0.267	0.267
		(0.101)	(0.535)	(0.535)
Team Size		0.003	-0.056	-0.056
		(0.110)	(0.524)	(0.524)
Constant	0.650***	0.496	0.132	0.132
	(0.110)	(0.638)	(3.001)	(3.001)
R^2	0.0198	0.0923		
Log Likelihood	-23.49	-22.03	-21.01	-21.01
Pseudo R^2			0.0810	0.0810
Estimation	OLS	OLS	Logit	Logit
Observations	38	38	38	38

Table A.4: Primary Field Study: Regression Analysis of Firm Outcomes Interacted with Firm Characteristics. Ordinary least squares (OLS) estimation of cross-sectional firm-level data. We introduce firm characteristics of *Firm Size* and *Graduate Degree* as interaction terms; both variables are demeaned to facilitate interpretation of the interaction terms, i.e., the incremental effect of the experimental treatment of *Firm Size* for one additional person, at the mean *Firm Size*. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	Value		Novelty		Completion		Evaluation	
Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Treatment Group	0.603*	0.526*	-0.548†	-0.817*	0.261	0.210	0.143	0.157
	(0.259)	(0.227)	(0.276)	(0.330)	(0.350)	(0.484)	(0.167)	(0.190)
Firm Size	-0.069	-0.209	-0.303	-0.772*	-0.456†	-0.606	0.260†	0.290
	(0.305)	(0.208)	(0.225)	(0.320)	(0.236)	(0.375)	(0.148)	(0.202)
Treatment Group	0.102	0.154	0.385	0.878*	0.792*	0.882†	-0.430†	-0.439
× Firm Size	(0.355)	(0.224)	(0.376)	(0.317)	(0.350)	(0.477)	(0.216)	(0.266)
Graduate Degree	-0.166	0.301	-0.515	0.030	-1.345*	-1.133	-0.085	-0.236
	(0.463)	(0.552)	(0.354)	(0.674)	(0.510)	(0.761)	(0.319)	(0.337)
Treatment Group	0.474	0.256	-0.035	-0.775	0.530	0.372	0.182	0.275
× Graduate Degree	(0.718)	(0.603)	(0.683)	(0.843)	(0.883)	(1.022)	(0.443)	(0.471)
Current Student		0.689†		0.114		0.298		-0.228
		(0.363)		(0.371)		(0.601)		(0.331)
Github		0.117		1.330		-0.074		-0.097
		(0.920)		(1.027)		(1.222)		(0.312)
Google Development		-0.152		1.332*		0.159		0.012
		(0.394)		(0.537)		(0.702)		(0.444)
Software Development		-0.050		0.002		-0.039		0.008
		(0.030)		(0.035)		(0.045)		(0.025)
Prior Hackathons		-0.144		-0.061		-0.076		0.033
		(0.089)		(0.127)		(0.134)		(0.093)
No Evaluation	-3.506***	-3.734***	-3.467***	-3.593***	-3.008***	-3.142***		
	(0.245)	(0.209)	(0.223)	(0.270)	(0.315)	(0.378)		
Constant	3.291***	3.440***	3.625***	2.082*	2.838***	3.028*	0.590***	0.699*
	(0.284)	(0.866)	(0.189)	(0.904)	(0.316)	(1.187)	(0.118)	(0.337)
R^2	0.877	0.924	0.793	0.853	0.704	0.725	0.153	0.206
Sample	Full	Full	Full	Full	Full	Full	Full	Full
Observations	38	38	38	38	38	38	38	38

Table A.5: Primary Field Study: Variable Definitions and Summary Statistics of Firm Process from Source Code File Hierarchies. Alternate dependent variables constructed from source code file hierarchies over time defined below with their conceptual interpretation. Observations are at the firm-minute level, with 20,520 firm-minute observations across 38 firms.

Variable (Interp.)	Definition	Mean	SD	Min	Max
System-Level Branching (Integration)	Branching factor at the root of the file hierarchy. Equivalent to files and directories directly below the root.	6.271	6.021	1	56
Subsystem-Level Branching (Specialization)	Average branching factor for directories below the root of the file hierarchy.	2.271	1.624	1	7.667

Table A.6: Primary Field Study: Regression Analysis of Firm Processes from Source Code File Hierarchies. Ordinary least squares (OLS) estimation of firm-minute level data. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	System-Level Branching	Subsystem-Level Branching
Treatment x Post	3.258*	-0.981*
	(1.413)	(0.426)
Firm FE	Yes	Yes
Time FE	Yes	Yes
R^2	0.410	0.416
Adjusted R^2	0.394	0.400
Firms	38	38
Observations	20,520	20,520
Level	Firm-Minute	Firm-Minute

Table A.7: Primary Field Study: Firm Process Correlations. This table shows pairwise correlations for the dependent variables used in the firm-minute level panel analysis of organizational search processes.

		(1)	(2)	(3)	(4)
(1)	System-Level Branching	1			
(2)	Subsystem-Level Branching	0.320	1		
(3)	Code Integration Action	0.517	0.385	1	
(4)	Advanced API	0.220	0.302	0.258	1

Table A.8: Primary Field Study: Regression Analysis of Firm Process at Firm-Post Level. This analysis of firm search process resembles the main analysis of the paper, except that firm-minute observations are aggregated into a single observation in the pre-treatment period and a single observation in the post-treatment period. All dependent variables are averaged over each pre- and post-treatment period of time. Time fixed effects represent indicators for each hour of the experiment. Ordinary least squares (OLS) estimation of firm-minute level data. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	(1) System-Level Branching	(2) Subsystem-Level Branching	(3) Code Integration Action	(4) Advanced API
Treatment x Post	3.258*	-0.981*	2.074*	-1.124**
	(1.413)	(0.426)	(0.878)	(0.408)
Post	3.613***	1.831***	1.532***	1.497***
	(0.984)	(0.358)	(0.420)	(0.361)
Firm FE	Yes	Yes	Yes	Yes
Time FE	No	No	No	No
R^2	0.619	0.552	0.530	0.439
Adjusted R^2	0.609	0.540	0.517	0.424
Firms	38	38	38	38
Observations	76	76	76	76
Level	Firm-Pre/Post	Firm-Pre/Post	Firm-Pre/Post	Firm-Pre/Post

Table A.9: Primary Field Study: Regression Analysis of Firm Process Allowing for Time Heterogeneity. This table show the regression analysis using three separate Treatment × Post variables representing each time period after each of the three coordination exchanges that occurred in the treatment period. Ordinary least squares (OLS) estimation of firm-minute level data. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	(1) System-Level Branching	(2) Subsystem-Level Branching	(3) Code Integration Action	(4) Advanced API Specialization
First Treatment x Post	0.749	-0.738†	0.255	-0.429†
	(1.404)	(0.383)	(0.480)	(0.230)
Second Treatment x Post	3.008†	-1.123*	1.714†	-1.279*
	(1.593)	(0.482)	(0.917)	(0.484)
Third Treatment x Post	6.016***	-1.083*	4.255**	-1.663**
	(1.579)	(0.475)	(1.365)	(0.581)
Firm FE	Yes	Yes	Yes	Yes
Time FE	Yes	Yes	Yes	Yes
R^2	0.446	0.420	0.525	0.372
Adjusted R^2	0.431	0.404	0.512	0.355
Firms	38	38	38	38
Observations	20,520	20,520	20,520	20,520
Level	Firm-Minute	Firm-Minute	Firm-Minute	Firm-Minute

Table A.10: Primary Field Study: Regression Analysis of Firm Productivity. Using the same estimation method for Firm Search Process used in the main paper, we estimate the effect of our experimental treatment on the general productivity of the firms, measured in terms of *Lines*, representing the overall lines of software code written by the team, and ln(Lines + 1), the natural log of the same variable. Ordinary least squares (OLS) estimation of firm-minute level data. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	(1) Lines	(2) ln(Lines + 1)
Treatment x Post	31155.012	-0.403
	(36138.152)	(0.791)
Firm FE	Yes	Yes
Time FE	Yes	Yes
R^2	0.0993	0.592
Adjusted R^2	0.0750	0.581
Firms	38	38
Observations	20,520	20,520
Level	Firm-Minute	Firm-Minute

Table A.11: Primary Field Study: Regression Analysis of Firm Process Controlling for Firm Productivity. This table shows the same regression models for Firm Search Process as in the main paper, except including *ln(Lines + 1)* as a control for time-variant heterogeneity in firm productivity in generating software code, measured in terms of the logarithm of the lines of software code written by the firm. Ordinary least squares (OLS) estimation of firm-minute level data. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	(1) System-Level Branching	(2) Subsystem-Level Branching	(3) Code Integration Action	(4) Advanced API Specialization
Treatment x Post	3.653**	-0.892*	2.163*	-1.072**
	(1.139)	(0.382)	(0.845)	(0.386)
Ln(Lines + 1)	0.980***	0.222***	0.220**	0.128**
	(0.174)	(0.058)	(0.071)	(0.043)
Firm FE	Yes	Yes	Yes	Yes
Time FE	Yes	Yes	Yes	Yes
R^2	0.560	0.541	0.481	0.381
Adjusted R^2	0.549	0.529	0.467	0.364
Firms	38	38	38	38
Observations	20,520	20,520	20,520	20,520
Level	Firm-Minute	Firm-Minute	Firm-Minute	Firm-Minute

Table A.12: Primary Field Study: Comparison of Meeting Duration & Post-Meeting Latency. The rows correspond to the sum of meeting duration and post-meeting duration for the three mentor meetings, with a stand-up meeting in treatment, and the total across all the meetings. The first three columns show the mean and standard deviation (in parentheses) for control, treatment, and the full sample. The last column shows the differences between treatment and control samples and the associated standard error (in parentheses). A t-test of difference in means finds no statistically significant difference between the treatment and control samples for any meeting or the total.

Meeting Duration	Sample			
& Post-Meeting Latency	Control	Treatment	Full	Difference
Meeting 1	13.70	19.00	16.21	5.300
	(23.56)	(23.89)	(23.55)	(7.706)
Meeting 2	14.50	9.944	12.34	-4.556
	(30.01)	(24.47)	(27.26)	(8.947)
Meeting 3	19.30	20.06	19.66	0.756
	(30.85)	(35.24)	(32.55)	(10.72)
Total for All Meetings	47.50	49.00	48.21	1.500
	(51.89)	(56.77)	(53.52)	(17.63)

Table A.13: Primary Field Study: Regression Analysis of Firm Process Interacted with Firm Characteristics. Ordinary least squares (OLS) estimation of firm-minute level data. We introduce firm characteristics of *Firm Size* and *Graduate Degree* as interaction terms; their baseline values drop out of the regression because they are time-invariant and collinear with the firm fixed effects. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	System-Level Branching			Subsystem-Level Branching		
Variable	(1)	(2)	(3)	(4)	(5)	(6)
Treatment × Post	2.729	2.981†	2.059	-2.116*	-1.175**	-2.740*
	(3.048)	(1.724)	(3.560)	(0.914)	(0.421)	(1.041)
Treatment × Post	0.194		0.308	0.417		0.523
× Firm Size	(0.946)		(0.980)	(0.350)		(0.354)
Treatment × Post		0.737	0.961		0.516	0.895
× Graduate Degree		(2.927)	(3.037)		(0.520)	(0.538)
Firm FE	Yes	Yes	Yes	Yes	Yes	Yes
Time FE	Yes	Yes	Yes	Yes	Yes	Yes
R^2	0.410	0.410	0.411	0.425	0.418	0.431
Adjusted R^2	0.394	0.394	0.395	0.409	0.402	0.415
Firms	38	38	38	38	38	38
Observations	20,520	20,520	20,520	20,520	20,520	20,520
Level	Firm-Min.	Firm-Min.	Firm-Min.	Firm-Min.	Firm-Min.	Firm-Min.

	Code Integration Action			Advanced API Specialization		
Variable	(7)	(8)	(9)	(10)	(11)	(12)
Treatment × Post	-5.454*	2.652*	-5.813*	-1.836*	-0.717	-1.154
	(2.126)	(1.175)	(2.626)	(0.840)	(0.524)	(0.862)
Treatment × Post	2.766**		2.826**	0.262		0.146
× Firm Size	(0.900)		(0.940)	(0.306)		(0.276)
Treatment × Post		-1.539	0.515		-1.084†	-0.978†
× Graduate Degree		(1.901)	(1.717)		(0.595)	(0.558)
Firm FE	Yes	Yes	Yes	Yes	Yes	Yes
Time FE	Yes	Yes	Yes	Yes	Yes	Yes
R^2	0.535	0.460	0.536	0.339	0.346	0.347
Adjusted R^2	0.523	0.445	0.523	0.321	0.328	0.329
Firms	38	38	38	38	38	38
Observations	20,520	20,520	20,520	20,520	20,520	20,520
Level	Firm-Min.	Firm-Min.	Firm-Min.	Firm-Min.	Firm-Min.	Firm-Min.

Table A.14: Primary Field Study: Mediation Analysis. Generalized structural equation model (SEM) estimation on cross-sectional data at the firm level. All models are estimated in one structural model. *Code Integration Action* and *Advanced API Specialization* serve as mediators for both $Value$ and $Novelty$. Robust standard errors clustered at the firm level shown in parentheses, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

Variable	Code Integration Action	Advanced API Specialization	Value	Novelty
Treatment	5.312***	-1.898***	-0.277	0.506
	(1.410)	(0.629)	(0.465)	(0.499)
Code Integration Action			0.147***	-0.030
			(0.047)	(0.031)
Advanced API Specialization			-0.093	0.404***
			(0.081)	(0.122)
Constant	1.828	1.043	3.906***	1.351
	(2.350)	(1.637)	(1.191)	(0.908)
Control Variables	Yes	Yes	Yes	Yes
Observations	38	38	38	38

Table A.15: Follow-On Laboratory Study: Background Characteristics of Study Participants. This table presents the means, standard deviations (in parentheses), and the F-statistic from an Analysis of Variance (ANOVA) for the background characteristics of the members of the teams. p-values for the F-test are shown in brackets. We do not find a statistically significant difference across these three groups in any of these background characteristics, verifying the validity of the experimental randomization. We average across the members of each team to form one observation per team. *Age* is measured in years. *Gender* indicates whether the subject is female (1) or male (0). *Graduate Education* indicates whether the individual has any graduate education. *Current Student* indicates whether the individual is a current student. *Any Experience* indicates whether the individual has any past experience in product development.

	Sample				
Variable	Condition 1	Condition 2	Condition 3	Total	F-Statistic
Age	35.35	35.86	33.67	34.97	0.627
	(7.567)	(6.836)	(6.561)	(6.960)	[0.537]
Gender	0.464	0.458	0.507	0.476	0.190
	(0.297)	(0.292)	(0.299)	(0.293)	[0.827]
Graduate Education	0.478	0.486	0.435	0.467	0.172
	(0.331)	(0.326)	(0.309)	(0.318)	[0.843]
Current Student	0.348	0.431	0.333	0.371	0.786
	(0.309)	(0.269)	(0.284)	(0.287)	[0.460]
Any Experience	0.203	0.236	0.145	0.195	0.682
	(0.280)	(0.269)	(0.263)	(0.269)	[0.509]

Table A.16: Follow-On Laboratory Study: Additional Variable Definitions and Sources. Additional measures of outcomes, process, intervention duration, and ad hoc communication drawn from each team's *Final Output* design, *Video Recording* of their working session, their *Individual Sketches*, and a *Post Survey* after the experiment.

Variable	Definition	Source
Outcomes		
Completeness	Completeness of the final product (Likert 1–5). Average of two independent rater assessments.	Final Output
Process		
Coordination (MULTIPLE)	Self-reported responses to five survey measures of coordination performance based on Lewis (2003) (Likert 1–5): *Effectiveness*, *Few Misunderstandings*, *Low Backtracking*, *Efficiency*, and *Low Confusion*.	Post Survey
Specialization (MULTIPLE)	Self-reported responses to five survey measures of individual specialization performance based on Lewis (2003) (Likert 1–5): *Group*, *Individual*, *Responsibility*, *Necessity*, and *Awareness*.	Post Survey
Intervention Duration		
Meeting Duration	Duration in seconds of all formal meetings associated with experimental intervention(s), measured from the time the mentor enters the room to when he exits, which is after the last statement by any member in response to a intervention question.	Video Recording
Post-Meeting Latency	Duration in seconds of the total time after the mentor exits the room and until the team begins working on either individual sketches or engages in ad hoc communication related to the product.	Video Recording
Ad Hoc Communication		
Ad Hoc Frequency	Count of distinct exchanges not including those related to the experimental intervention, where each exchange is an individual's oral communication bounded by speech by other individuals on the team.	Video Recording
Ad Hoc Word Count	Count of words spoken by anyone on the team not including those related to the experimental intervention.	Video Recording

Table A.17: Follow-On Laboratory Study: Additional Variables Summary Statistics and Cross-Sectional Analysis. The first three columns contain the mean and in parentheses the standard deviation of teams in each condition. The last two columns compare Conditions 1 vs. 2 and Conditions 2 vs. 3, respectively, based on a t-test of the difference in means; the values reflect the difference in means and in parentheses the standard error, with significance indicated by $^{\dagger}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$.

	Sample				Difference in Means	
	Full	Condition 1	Condition 2	Condition 3	1 vs. 2	2 vs. 3
Outcomes						
Completeness	3.381	3.362	3.389	3.391	0.0266	0.00242
	(0.675)	(0.778)	(0.753)	(0.478)	(0.223)	(0.185)
Process						
Coor.: Effectiveness	4.133	3.580	4.181	4.638	0.601**	0.457**
	(0.719)	(0.760)	(0.461)	(0.481)	(0.182)	(0.137)
Coor.: Few Misund.	4.062	3.623	4.083	4.478	0.460†	0.395†
	(0.872)	(0.895)	(0.806)	(0.724)	(0.248)	(0.224)
Coor.: Low Bktrk.	2.705	2.319	2.694	3.101	0.376	0.407†
	(0.800)	(0.693)	(0.839)	(0.685)	(0.225)	(0.224)
Coor.: Efficiency	4.095	3.913	4.153	4.217	0.240	0.0646
	(0.817)	(1.065)	(0.674)	(0.656)	(0.259)	(0.194)
Coor.: Low Confsn.	2.862	2.638	2.764	3.188	0.126	0.425†
	(0.734)	(0.643)	(0.813)	(0.642)	(0.214)	(0.214)
Spec.: Group	3.733	4.000	3.722	3.478	-0.278†	-0.244
	(0.656)	(0.522)	(0.570)	(0.771)	(0.160)	(0.197)
Spec.: Individual	3.248	3.754	3.264	2.725	-0.490*	-0.539*
	(0.891)	(0.812)	(0.792)	(0.789)	(0.234)	(0.231)
Spec.: Responsibility	3.271	3.333	3.278	3.203	-0.0556	-0.0749
	(0.736)	(0.876)	(0.570)	(0.764)	(0.215)	(0.196)
Spec.: Necessity	3.776	3.841	3.778	3.710	-0.0628	-0.0676
	(0.780)	(0.828)	(0.727)	(0.812)	(0.227)	(0.225)
Spec.: Awareness	3.414	3.406	3.361	3.478	-0.0447	0.117
	(0.726)	(0.531)	(0.589)	(0.999)	(0.164)	(0.238)
Intervention Duration						
Meeting Duration	101.0	68.83	98.63	135.6	29.80*	36.98*
	(53.10)	(36.94)	(50.46)	(49.76)	(12.95)	(14.63)
Post-Meeting Latency	38.61	32.74	31.25	52.17	-1.489	20.92*
	(30.42)	(26.46)	(26.70)	(34.19)	(7.757)	(8.927)
Ad Hoc Communication						
Ad Hoc Frequency	162.2	127.4	161.8	197.3	34.44†	35.43†
	(68.17)	(52.79)	(78.39)	(53.02)	(19.58)	(19.61)
Ad Hoc Word Count	5749.8	5611.3	5795	5841.1	183.7	46.09
	(1774.7)	(1630.0)	(1833.7)	(1917.1)	(506.9)	(547.1)

Table A.18: Follow-On Laboratory Study: Survey-Based Measures of Coordination and Specialization. These survey measures come from Lewis (2003) and subsequent work.

Variable	Survey Question
Coordination	
Effectiveness	"Our team worked together in a well-coordinated fashion."
Few Misunderstandings	"Our team had very few misunderstandings about what to do."
Low Backtracking	"Our team needed to backtrack and start over a lot." (Reversed)
Efficiency	"We accomplished the task smoothly and efficiently."
Low Confusion	"There was much confusion about how we would accomplish the task." (Reversed)
Specialization	
Group	"Each team member has specialized knowledge of some aspect of our project."
Individual	"I have knowledge about an aspect of the project that no other team member has."
Responsibility	"Different team members are responsible for expertise in different areas."
Necessity	"The specialized knowledge of several different team members was needed to complete the project deliverables."
Awareness	"I know which team members have expertise in specific areas."

Table A.19: Follow-On Laboratory Study: Correlations. This table shows pairwise correlations for the dependent variables used in the follow-on laboratory study.

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1)	*Value*	1									
(2)	*Novelty*	0.018	1								
(3)	*Completeness*	0.287	0.460	1							
(4)	*Time to Integrate*	-0.140	0.169	-0.012	1						
(5)	*Individual Sketches*	-0.274	0.151	0.065	0.483	1					
(6)	*Coor.: Effectiveness*	0.352	-0.194	0.053	-0.407	-0.280	1				
(7)	*Coor.: Few Misund.*	0.395	-0.105	0.102	-0.389	-0.145	0.734	1			
(8)	*Coor.: Low Bktrk.*	0.370	-0.156	0.110	-0.056	-0.135	0.563	0.474	1		
(9)	*Coor.: Efficiency*	0.114	-0.158	0.044	-0.154	0.066	0.699	0.654	0.495	1	
(10)	*Coor.: Low Confsn.*	0.326	0.002	0.208	-0.092	-0.054	0.551	0.486	0.793	0.438	1
(11)	*Spec.: Group*	-0.311	-0.046	-0.102	0.125	0.227	0.097	0.159	0.041	0.274	0.059
(12)	*Spec.: Individual*	-0.429	0.045	-0.181	0.067	0.203	-0.284	-0.140	-0.355	-0.057	-0.223
(13)	*Spec.: Responsibility*	-0.192	-0.090	-0.120	-0.083	-0.111	0.135	0.089	-0.130	0.141	-0.123
(14)	*Spec.: Necessity*	-0.177	-0.081	-0.059	-0.033	0.020	0.255	0.141	-0.043	0.120	-0.066
(15)	*Spec.: Awareness*	-0.168	-0.150	-0.130	-0.037	-0.194	0.241	0.251	0.128	0.204	0.130
(16)	*Meeting Duration*	0.447	-0.105	0.052	-0.069	-0.390	0.213	0.204	0.190	-0.103	0.071
(17)	*Post-Meeting Latency*	0.174	-0.033	0.017	-0.008	-0.162	0.225	0.110	0.127	0.013	0.037
(18)	*Ad Hoc Frequency*	0.175	-0.012	0.076	-0.030	-0.166	0.263	0.257	0.212	0.072	0.196
(19)	*Ad Hoc Word Count*	-0.062	0.084	-0.055	0.069	-0.037	-0.157	-0.206	0.004	-0.358	-0.052

		(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
(11)	*Spec.: Group*	1								
(12)	*Spec.: Individual*	0.420	1							
(13)	*Spec.: Responsibility*	0.602	0.456	1						
(14)	*Spec.: Necessity*	0.540	0.301	0.596	1					
(15)	*Spec.: Awareness*	0.625	0.419	0.582	0.525	1				
(16)	*Meeting Duration*	-0.240	-0.289	-0.045	-0.143	0.195	1			
(17)	*Post-Meeting Latency*	-0.171	-0.307	-0.030	-0.011	0.003	0.449	1		
(18)	*Ad Hoc Frequency*	-0.097	-0.255	0.068	-0.082	0.095	0.413	0.044	1	
(19)	*Ad Hoc Word Count*	-0.333	-0.038	-0.212	-0.092	-0.142	0.193	0.030	0.249	1