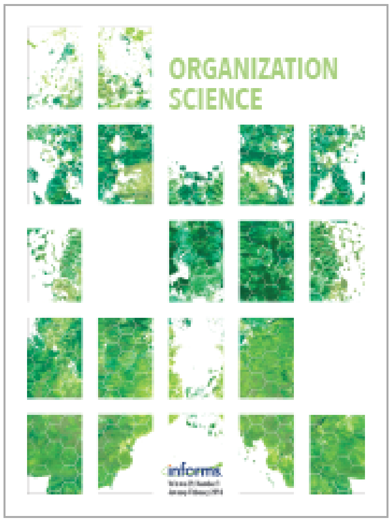
This article was downloaded by: [114.214.215.95] On: 15 February 2023, At: 00:22
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA



Organization Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Social Exchange and the Reciprocity Roller Coaster: Evidence from the Life and Death of Virtual Teams

Jérôme Hergueux, Emeric Henry, Yochai Benkler, Yann Algan

To cite this article:
Jérôme Hergueux, Emeric Henry, Yochai Benkler, Yann Algan (2021) Social Exchange and the Reciprocity Roller Coaster: Evidence from the Life and Death of Virtual Teams. Organization Science

Published online in Articles in Advance 29 Oct 2021

. https://doi.org/10.1287/orsc.2021.1515

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2021, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

informs
http://pubsonline.informs.org/journal/orsc
ORGANIZATION SCIENCE
Articles in Advance, pp. 1–19
ISSN 1047-7039 (print), ISSN 1526-5455 (online)

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Social Exchange and the Reciprocity Roller Coaster: Evidence from the Life and Death of Virtual Teams

Jérôme Hergueux,[a,b] Emeric Henry,[c] Yochai Benkler,[b,d] Yann Algan[c]

[a] French National Center for Scientific Research (CNRS, BETA Laboratory), 67000 Strasbourg, Alsace, France; [b] Berkman Klein Center for Internet & Society at Harvard University, Cambridge, Massachusetts 02138; [c] Department of Economics, Sciences Po, 75007 Paris, France; [d] Harvard Law School, Cambridge, Massachusetts 02138

Contact: jerome.hergueux@gess.ethz.ch, https://orcid.org/0000-0003-0294-7938 (JH); emeric.henry@sciencespo.fr, https://orcid.org/0000-0001-9851-084X (EH); ybenkler@law.harvard.edu, https://orcid.org/0000-0001-9593-9173 (YB); yann.algan@sciencespo.fr, https://orcid.org/0000-0002-8120-5969 (YA)

Received: July 30, 2019
Revised: May 30, 2020; August 11, 2020; February 25, 2021; April 19, 2021; July 9, 2021
Accepted: July 26, 2021
Published Online in Articles in Advance: October 29, 2021

https://doi.org/10.1287/orsc.2021.1515

Copyright: © 2021 INFORMS

Abstract. Organizations are riddled with cooperation problems, that is, instances in which workers need to *voluntarily* exert effort to achieve efficient collective outcomes. To sustain high levels of cooperation, the experimental literature demonstrates the centrality of reciprocal preferences but has also overlooked some of its negative consequences. In this paper, we ran lab-in-the-field experiments in the context of open-source software development teams to provide the first field evidence that highly reciprocating groups are not necessarily more successful in practice. Instead, the relationship between high reciprocity and performance can be more accurately described as U-shaped. Highly reciprocal teams are generally more likely to fail and only outperform other teams conditional on survival. We use the dynamic structure of our data on field contributions to demonstrate the underlying theoretical mechanism. Reciprocal preferences work as a catalyst at the team level: they reinforce the cooperative equilibrium in good times but also make it harder to recover from a negative signal (the project dies). Our results call into question the idea that strong reciprocity can shield organizations from cooperation breakdowns. Instead, cooperation needs to be dynamically managed through relational contracts.

History: This paper has been accepted for the *Organization Science* Special Issue on Experiments in Organizational Theory.

Funding: This research was funded by the consolidator grant agreement [Grant 647870] from the European Research Council, European Union's Horizon 2020 Research and Innovation Programme (Yann Algan).

Supplemental Material: The online appendix is available at https://doi.org/10.1287/orsc.2021.1515.

Keywords: cooperation • reciprocity • social exchange • organizational behavior • virtual teams • open-source software

1. Introduction

Public goods problems are ubiquitous within organizations.[1] Because employment contracts cannot enumerate (or even predict) the myriad cooperation opportunities that may arise within organizations, workers often need to cooperate *voluntarily* (i.e., at a cost to themselves) in order to achieve efficient collective outcomes. In most industries, successful organizations are the ones in which individuals readily cooperate by sharing relevant information and knowledge, helping each other and collectively engaging in problem solving.[2]

Over the past decades, however, experimental research has shown that cooperative equilibria are typically unsustainable within organizations (Ledyard 1995, Chaudhuri 2011). The reason for this is that most individuals endorse the norm of reciprocity in social dilemma situations (Gouldner 1960, Dufwenberg and Kirchsteiger 2004, Sobel 2005). Reciprocal individuals behave as "conditional cooperators": they are willing to cooperate as long as they think that others will respond in kind (see Chaudhuri 2011 for a survey). A major consequence of this preference is that cooperation is highly fragile within organizations (Fischbacher et al. 2001, Fischbacher and Gächter 2010). Reciprocal workers withdraw their willingness to cooperate whenever they feel like their coworkers are not doing their "fair share." In repeated experiments, this triggers a self-reinforcing process that eventually leads to the breakdown of cooperation (see Ledyard 1995 for an extensive review).

To sustain cooperation at the group level, experimental research has identified two classes of mechanisms aimed at enforcing strong reciprocity: decentralized monetary punishment (Fehr and Gächter 2000a, Gächter et al. 2008) and screening through peer selection or fiat (Cinyabuguma et al. 2005, Page et al. 2005, Charness and Yang 2014). The message of this literature is that the cooperation breakdown problem can be solved (i.e., a cooperative equilibrium

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

reached) by mechanisms that enable strong reciprocity at the group level, such as designing employment contracts that screen nonreciprocal workers (Kosfeld and von Siemens 2011, Bartling et al. 2012).

In this paper, we argue that strong reciprocity does not shield organizations from the cooperation breakdown problem. The reason is quite simple: to the contrary of most laboratory experiments,[3] the details of the cooperative game played by organization members are usually not common knowledge. In the field, individuals need to learn the details of the cooperative game they play as they work together. This includes the set of possible actions available to each player in various (and possibly unforeseen) contingencies together with their associated costs and organizational benefits. Those parameters are typically heterogeneous across group members and subject to change over time.

The consequence is straightforward: individuals in the field receive noisy signals of the willingness of their peers to cooperate. Those signals must then be interpreted, notably in light of the history of previous interactions within the organization—otherwise referred to as "norms" or "culture" (Kreps et al. 1996). Because reciprocity is both the key to sustained cooperation and the main mechanism through which it unravels (Fischbacher and Gächter 2010, Chaudhuri 2011), highly reciprocating groups may not always be more successful in practice. Over the course of the relationship, even highly reciprocating teams may receive negative signals that can lead to the unraveling of cooperation. In other words, strong reciprocal preferences allow organizations to sustain efficient cooperative equilibria, leading to above-average performance, but they also make them *more* sensitive to the cooperation breakdown problem (in the case of bad news).

This paper tests these ideas in the field context of Sourceforge, a large open-innovation platform hosting virtual teams that seek to develop novel open-source software (OSS) products. This environment is particularly well suited to our research purposes. OSS projects tend to be complex and uncertain and largely rely on voluntary code contributions from team members in order to thrive. At the same time, the resulting software product is freely available for anyone to use. Team members, therefore, face a repeated public goods dilemma in which they receive (noisy) signals that reflect their teammates' past contributions and have to decide how much they want to contribute to the project, if at all, in the current period.

In 2011, we contacted a sample of 2,534 open-source developers registered with Sourceforge to participate in an online experiment. We collected laboratory data on 1,194 subjects (i.e., a 47% participation rate) and used their conditional contribution decisions in a public goods game to construct a laboratory measure of reciprocal preferences. In parallel, we collected field data on all the projects to which our experimental subjects *and* their collaborators had ever contributed. This yielded a final sample of 5,557 projects, many of which had failed, and involving 10,537 developers overall. For each project, we extracted the monthly number of code contributions (or "commits") made by each developer.

An important feature of field research is that it is only possible to recruit subjects from active organizations at the time of the experiment. In order to expand our research sample in the direction of previously failed projects and their members, we use our laboratory measure of reciprocity to validate a generalizable measure of reciprocity computed on the basis of the historical dynamics of project contributions between developers. We then use this field measure to expand our research sample in the direction of previously failed projects and their members. This strategy allows us to separately analyze the determinants of failure (i.e., the project died) from that of performance (measured through Sourceforge's flagship indicator).

The paper makes two main contributions to the literature on cooperation. First, we show that organizations with a higher share of reciprocal members are not necessarily more successful. Instead, the relationship between high reciprocity and success can be more accurately described as U-shaped. Highly reciprocal teams are significantly more likely to fail and only outperform other teams conditional on survival. Second, we analyze the dynamics of field contributions in the data to identify the underlying microlevel mechanism. Reciprocal preferences work as a catalyst at the organizational level. They reinforce the cooperative equilibrium (foster increased contribution levels) when team members receive positive signals. However, they also accelerate the slowdown when they receive negative ones, leading to a lower probability of recovery after a period of inactivity (the project fails).

Our field results suggest that strategies aimed at screening workers according to their social type (through, e.g., sorting (Page et al. 2005) or contract design (Bartling et al. 2012)) are unlikely to generate a self-sustained cooperative equilibrium within organizations. Instead, future experimental and theoretical research on cooperation should focus on how organizations can dynamically sustain a cooperative equilibrium in the face of adverse shocks and imperfect monitoring. To do this, they must define the implicit norms that govern behavior among team members because those implicit norms, or "relational contracts," allow workers to make sense of the signals they receive and, ultimately, to avoid inefficient cooperation breakdowns (Chassang 2010, Acemoglu and Jackson 2015).

2. Related Literature

2.1. The Lab-in-the-Field Literature

The early literature, which attempted to identify individual preferences, beliefs, and institutional designs

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

that encourage voluntary cooperation, relies on laboratory experiments (Ledyard 1995, Roth 1995). One of its most robust findings is that cooperation gradually decays in repeated experiments. Although most individuals initially make nonzero contributions, their willingness to cooperate steadily declines with repetition (see Chaudhuri 2011 for a survey). This finding follows from the fact that many individuals exhibit reciprocal preferences: they are conditional cooperators (i.e., willing to cooperate as long as others do so as well), hence, the positive initial contribution levels. However, because reciprocators have a preference for matching the contributions they observe or expect from others, cooperation inevitably collapses if (i) the group contains free-riders, who never cooperate in order to maximize their private payoffs; (ii) the group contains "weak" reciprocators, who behave as conditional cooperators but with a "self-serving bias" (Fischbacher et al. 2001) (i.e., they contribute but generally less than others); and (iii) some group members have reasons to believe that others will reduce their contributions in the future.

The major takeaway from this laboratory-based literature is that, even though most individuals are not selfish, cooperation is highly fragile (Fischbacher and Gächter 2010). In particular, reciprocal preferences are crucial to sustain cooperation within organizations but require institutions that either discipline nonreciprocating types (through monetary and nonmonetary punishment) or provide a mechanism for excluding them from the group (Fehr and Gächter 2000b, Chaudhuri 2011, Bartling et al. 2012).

More recently, the literature has questioned the field (referred to as "external" or "ecological") validity of these laboratory-based results. Because reciprocal preferences are hard to identify in the wild, empirical papers increasingly rely on "lab-in-the-field" designs (see Gneezy and Imas 2017 for a detailed description). This fast-growing literature relies on validated laboratory paradigms to elicit psychological traits or preferences and links these measures to outcomes of theoretical interest in the field. In other words, the spirit of lab-in-the-field experiments is to match individuals' behaviors in experimental games (the laboratory) with choices by the same individuals in the field and/or the performance of the organizations in which they work. This approach is a powerful tool for organizational researchers interested in microfounding the processes that underlie aggregate outcomes in the field (Felin and Foss 2005, Felin et al. 2015, Bitektine et al. 2018).

Initial lab-in-the-field studies link behavior in the laboratory to individual behaviors and outcomes in field environments without focusing on group-level outcomes. In a seminal paper in economics, Karlan (2005) obtains experimental measures of reciprocity at the individual level and shows that they predict loan repayment among participants in a microcredit program. Charness and Villeval (2009) show that senior workers are typically more cooperative than junior ones in a standard public goods game. Fehr and Leibbrandt (2011) and Leibbrandt (2012) conduct a public goods game among Brazilian shrimp catchers and sellers and show that more cooperative subjects are less likely to engage in overextraction and achieve better market outcomes. Kosfeld and Rustagi (2015) show that the way traditional leaders "punish" players in their community based on how they behave in a public goods game predicts how successfully they cooperate in the field.[4]

However, there are fewer lab-in-the-field studies devoted to the link between reciprocal preferences and group-level outcomes. The first paper to study this question in a field setting is Anthony (2005), who does not adopt a lab-in-the-field approach per se, but instead relies on survey answers. The paper measures reciprocal behavior in 106 microcredit borrowing groups in the United States through a number of survey questions answered by randomly selected group members. It finds reciprocity to be the variable most closely associated with low levels of loan delinquency and higher group longevity. Barr and Serneels (2009) are the first to obtain experimental measures of reciprocal preferences from a sample of workers in 20 Ghanaian manufacturing firms, which they then couple with survey-based data on workers' individual wages and aggregate firm productivity. They find that reciprocal workers generally earn higher wages and report a strong firm-level relationship between reciprocating behavior and aggregate productivity. Rustagi et al. (2010) study 49 local groups that participated in a publicly funded forest conservation program in Ethiopia, where they were responsible for maintaining and cultivating the forest (the "public good"). They elicit reciprocal preferences using a conditional public goods game (Fischbacher et al. 2001) and notably measure the share of conditional and weakly conditional cooperators in each group, which they then associate with an independently collected measure of success in forest commons management. They find that groups with a larger share of highly reciprocating types are generally more successful. Similarly, Carpenter and Seki (2011) use the public goods game to elicit reciprocal preferences from Japanese fishermen. Even though their sample only contains 12 fishing crews, they find that those that exhibit higher levels of reciprocity are generally more productive.

Taken together, those papers reveal a clear empirical result: groups composed of more reciprocal types, as measured by incentivized laboratory experiments, achieve significantly better outcomes. The evidence

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

comes from a variety of field settings in which members face social dilemmas, such as forest commons management, profit-maximizing firms, and financial markets. This unambiguous relationship between reciprocity and group success is surprising in light of the laboratory literature on social dilemma. In this literature, reciprocal preferences drive cooperation levels down whenever conditional cooperators observe (or believe) that some team members are not doing their fair share.

We add to this literature by highlighting a potential limitation of the lab-in-the-field methodology, which may mechanically produce the strong positive relationship observed between reciprocity and organizational success. The key to lab-in-the-field designs is to recruit the experimental subjects directly from their work environment. We argue that this introduces a bias at both the individual and organizational levels. Failed organizations are much less likely to be sampled because the members are no longer present on the research site. More generally, for organizations that still exist, the propensity of members to participate in the experiment might be strongly correlated with the level of activity of the project. This sampling bias is likely to be particularly severe in field settings in which an inability to sustain cooperation can more easily lead to the death of the organization (e.g., in competitive markets, innovation-driven activities, or volunteer communities).

To address this concern, we propose an additional way to use laboratory data in the field: we use the laboratory data as a benchmarking tool for subjects' reciprocity motives. We then propose a generalizable field measure of reciprocity (inspired by the experimental game) and show that the laboratory and field measures are strongly correlated. To build our field measure, we leverage the panel structure of our data and measure developers' reactions to the past observed contribution decisions of their teammates (see Section 4 for details). Because this field measure is computed based on archival data, we can use it to expand our research sample in the direction of previously failed organizations.

2.2. Reciprocity and Social Exchange Theory (SET)

At a theoretical level, our paper is related to the literature on SET. Since the early contributions of Homans (1961), Blau (1964), and Emerson (1976), SET has been a highly influential theoretical construct in social psychology and organizational behavior. More precisely, our experimental and field measures of reciprocity are directly linked to the most paradigmatic instance of social exchange, known as "reciprocal exchange" (Cropanzano and Mitchell 2005). In this framework, reciprocity can be seen as a social norm (Gouldner 1960) whereby an individual receiving an unconditional benefit from another party feels bound to respond in kind.[5] Positive reciprocity dynamics, therefore, lead to a form of mutual commitment resulting in a virtuous, self-reinforcing cycle of cooperation.

Much of the existing literature to date relies on laboratory experiments to test these predictions (see Cook and Emerson 1978, Montgomery 1996, Molm et al. 2000, Molm 2003 as well as Cook et al. 2013 for a review). Our results add to this literature by exploring how the dynamics of reciprocal exchange occur in an online generalized exchange system, such as OSS, in which peers intend to coproduce a public good based on voluntary contributions. We take advantage of the fact that OSS teams differ in the extent to which their members endorse reciprocity. This allows us to link group-level reciprocity to an objective measure of organizational success. By combining lab-in-the-field experiments with the analysis of behavioral data over time, we hope to convince organizational researchers that field research can achieve relatively high levels of internal validity as well as ecological relevance (Schram 2005). We argue that this is especially true in computerized environments in which researchers can gather a significant amount of data on field behavior (Lazer et al. 2009) while retaining the ability to conduct controlled experiments (Hergueux and Jacquemet 2015).

Note that our work should be set apart from the important literature on negative reciprocity. In a recent survey of the literature on social exchange, Cropanzano et al. (2017, p. 29) note that SET actually "fails to distinguish the presence of negative constructs (e.g., abuse) from the absence of positive constructs (e.g., support)." They argue that this leads to some confusion in terms of behavioral predictions: negative behavior is predicted to lead to negative reciprocal responses (i.e., negative reciprocity), whereas the *absence* of positive behavior should, instead, lead to a lack of positive response (i.e., the decay of cooperation). Our empirical results illustrate the relevance of this distinction for SET: in the face of relative inactivity, reciprocal developers adjust their own cooperation level downward and eventually stop contributing. Because the ability to punish nonreciprocating types is altogether absent from our field of study (see Section 3.1), our paper is not related to the literature on negative reciprocity. In this respect, our field setting is most closely related to laboratory designs in which subjects can select their teammates based on observed past behavior, which has been found to have a dramatic impact on their ability to sustain very high levels of cooperation over time (see Cinyabuguma et al. 2005, Page et al. 2005, Charness and Yang 2014).[6]

2.3. Relational Contracts

At a theoretical level, our paper also connects to a related but distinct literature on relational (or "implicit") contracts in organizational economics. Chassang (2010) studies how agents can develop a successful cooperative

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

relationship when the details of cooperation are not common knowledge. In his model, teammates can observe agents' actions and their associated payoffs. However, there is uncertainty regarding which actions are "cooperative" because they do not always yield the intended organizational benefits. Relationships are, therefore, quite sensitive to adverse shocks and may only become resilient after significant "relational knowledge" has been built among the parties involved. The concept of relational knowledge implies that a workable subset of cooperative actions could be identified and developed as a routine at the organizational level. Because this routine develops as a function of the agents' previous histories, random events occurring during the relationship can lead to unexpected cooperation breakdowns and/or have a lasting impact on the way players approach cooperation. In a related paper, Gibbons and Henderson (2012) argue that the relational contracts that sustain cooperation in the field have to solve the twin problem of credibility (i.e., the misalignment of incentives within the organization) and clarity (i.e., the uncertainty over which actions are collectively considered cooperative for a given agent in a given situation). Such informal cooperative agreements take time to develop and are highly path-dependent (see also Acemoglu and Jackson 2015).[7]

Qualitative research shows that organizations dedicate enormous resources to building and maintaining relational contracts (Mayer and Argyres 2004, Kellogg 2009). Unfortunately, the existing experimental literature provides very little guidance regarding the type of informal norms that are most likely to sustain an efficient cooperative equilibrium at the organizational level with two notable exceptions. Fudenberg et al. (2012) study a repeated public goods game in which intended actions are implemented with noise (i.e., uncertain intentions). They experimentally show that "lenient" (i.e., not retaliating after the first signal of defection) and "forgiving" (i.e., returning to cooperation after retaliation) strategies are most efficient in this case. More recently, Gibbons et al. (2020) study a repeated bilateral trade relationship in which the state of the world is subject to exogenous shocks (i.e., uncertain environment). They find that, in such environments, relational contracts based on general principles perform better than those that adhere to specific rules but that high-performing relational contracts are typically difficult to build. By empirically showing that screening workers according to their social type is unlikely to eliminate the cooperation breakdown problem, we hope to encourage more experimental research of this sort.

3. Setup and Data Collection

3.1. Open-Source Software

To set the stage, we provide some background information on open-source software. OSS currently mobilizes millions of loosely connected developers from around the world who self-organize in virtual teams to develop software products (Faraj et al. 2011, Levine and Prietula 2013). OSS is responsible for most of the basic utilities on which the internet runs (e.g., the Apache Web server), popular programming languages (e.g., Python, R), and programming environments (e.g., Eclipse). It also competes with many of its proprietary counterparts in the realm of end-user applications (e.g., Android), operating systems (e.g., Linux), and web browsers (e.g., Firefox). At present, most businesses and public organizations rely on OSS for their daily activities (Walli et al. 2005, Ghosh 2007, Greenstein and Nagle 2014).

Apart from the aforementioned projects, which are both very large and quite well known, hundreds of thousands of smaller-scale OSS projects are hosted by online platforms, such as Sourceforge, which was dominant at the time of our study, and, more recently, Github. These platforms provide developers with a set of free standard online tools for collaborative software development (e.g., a code-versioning system, a bug tracker). Any developer can initiate a software project on such platforms, and the source code of each project is readily available for anyone to see and modify. Projects are, therefore, developed in the context of geographically distributed virtual teams that coordinate their activities in the absence of formal leadership, prespecified design rules, or markets (Benkler 2002, von Hippel and von Krogh 2003, von Krogh and von Hippel 2006). Contributors typically resolve potential disagreements over future developments through discussion and, in some rare cases, through "forking," whereby some team members decide to split off and develop their own version of the project. As a result, OSS is usually seen as a "technical meritocracy" (Scacchi 2007) in which developers typically acquire influence by contributing elegant code that "just works" (Weber 2004, Marlow et al. 2013). Similar to fundamental research, OSS development has, thus, been modeled as an evolutionary learning process driven by peer criticism and error correction (Lee and Cole 2003).

Because developers need to invest time and effort contributing to projects that are made freely available for anyone to use, OSS has been described as a privately produced public good (O'Mahony 2003) in which developers reveal their code in the expectation that others will reciprocate (Maurer and Scotchmer 2006). About 50% of OSS developers are volunteers who only contribute in their free time, and the other half derives either direct or indirect revenue from their contributions (Hertel et al. 2003, Lakhani and Wolf 2005). In the latter case, the developer can be paid by a firm to dedicate working hours to a project that serves corporate goals (Dahlander and Magnusson 2005). Some innovation-heavy firms (e.g., Google) also allow their employees to

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

dedicate working hours to any project of their choosing on the assumption that developing OSS (i) allows them to acquire new skills and (ii) keeps them in touch with a fast-moving open-innovation community.

3.2. Collecting Laboratory Data

In May 2011, 2,534 OSS developers registered with Sourceforge.net were contacted and asked to participate in an online experiment that we describe in more detail in Section 3.2.2. The experimental platform remained active for 10 complete days, and 1,194 subjects—a 47% take-up rate—participated. Before describing the details of the experimental procedure, we begin by describing how the initial sample of 2,534 developers was selected out of the large Sourceforge community that counted 221,802 projects registered in 2010.

3.2.1. Experimental Sample Selection. To select the initial pool to be contacted, we set up a two-tier selection procedure: first selecting projects and then selecting individuals within these projects. To select the projects, we used two stratification variables: size of project and type of license as described in the following. There is great heterogeneity between Sourceforge projects in terms of the number of contributors, and previous research efforts have been somewhat biased toward a handful of large and highly successful projects (Crowston et al. 2012). To avoid this pitfall, the first stratification variable we considered was the project size, defined as the number of contributors. Second, following Belenzon and Schankerman (2015), who argue that reciprocal developers prefer restrictive project licenses, we used the variable "license restrictiveness" as an additional stratification criterion, making it more likely that we would include diverse cooperative types in our pool.[8]

Specifically, we extracted all the Sourceforge projects that were active in 2010, defined as having either solved a bug or added a feature in 2010. This yielded a sample of 1,242 active projects. Of the 8,858 developers who contributed to those active projects, we identified those who had some development activity in 2010. We then ordered projects according to their number of active developers and relied on Belenzon and Schankerman's (2015) classification of the 44 existing OSS license types to label their licensing terms as highly, moderately, or weakly restrictive.

Because there were only 83 projects with more than seven active contributors, we selected all of those projects, irrespective of their license terms. For all the projects with six or fewer active contributors, we chose to construct a sample containing an equal number of highly, moderately, and weakly restrictive licenses. For instance, out of the 365 projects that had only one active developer, 239 featured highly restrictive licenses, 57 featured moderately restrictive licenses, and 69 featured weakly restrictive licenses. We, thus, retained the 57 projects with moderately restrictive licenses and then randomly selected 57 projects from the pool of projects with both highly and weakly restrictive licenses. We ended up with a sample of 322 active projects, balanced in terms of both size and license restrictiveness. Table 1 lists the number of projects selected by order of size and license type.

For the 322 projects selected, we kept all 1,019 developers who were active in 2010. In addition, we also randomly selected three nonactive developers. We ended up with a sample of 2,534 Sourceforge developers eligible to participate in the experiment. Table 1 summarizes the selection procedure.

3.2.2. The Online Experiment. With the support of the Sourceforge platform, we collected the email addresses of all 2,534 selected developers and sent them individual invitations to participate in the experiment. By clicking on a link included in the invitation message, eligible developers were able to log into the system with their Sourceforge username, which allowed us to identify them and subsequently collect their entire history of contributions to OSS. Subjects were then redirected to the welcome screen of the experimental platform.

Our design strictly follows the internet-specific procedures detailed in Hergueux and Jacquemet (2015). (See Online Appendix A for further details.) The key experimental game we used to elicit reciprocal preferences is the one-shot public goods game. This game is played in groups of four players, each with an initial endowment of \$10. Group members need to decide how much to contribute to a common project. Each dollar invested in the common project produces \$1.6, which is then equally distributed among group members. Thus, a \$1 investment only yields a private return of \$0.4 but benefits all other members of the group. This design captures the social dilemma faced by open-source developers in the field: contributing code to OSS can be individually costly but is socially efficient. Specifically, for player i who makes a contribution $contrib_i$, the final private payoff is given by

$$\pi_i = 10 - contrib_i + 0.4\sum_{j=1}^{4} contrib_j.$$

Following the example of Fischbacher et al. (2001), we elicited two types of contribution decisions: first an unconditional contribution and then a conditional contribution. For the unconditional contribution, each subject had to decide on a contribution in the game described. For the conditional contribution, each subject determined an intended contribution for each possible value (0, 1, 2, … 10) of the average contribution of the three other members of the group. The

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Table 1. Constructing the Sample of Eligible Subjects

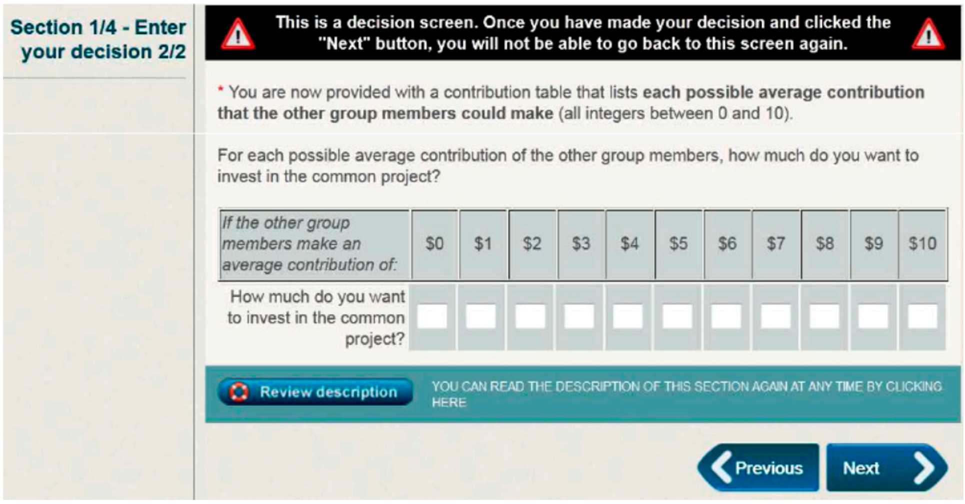
Number of active developers on project	License restrictiveness	Total number of projects	Number of projects randomly selected	Active developers randomly selected	Nonactive developers randomly selected
1	High	239	57	57	57
1	Moderate	57	57	57	57
1	Low	69	57	57	57
2	High	118	28	56	56
2	Moderate	28	28	56	56
2	Low	29	28	56	56
3	High	51	12	36	36
3	Moderate	12	12	36	36
3	Low	15	12	36	36
4	High	24	6	24	18
4	Moderate	13	6	24	18
4	Low	6	6	24	18
5	High	18	7	35	21
5	Moderate	9	7	35	21
5	Low	7	7	35	21
6	High	18	5	30	15
6	Moderate	5	5	30	15
6	Low	5	5	30	15
7+		83	83	962	249

conditional contributions allowed us to measure the subjects' willingness to behave reciprocally (i.e., to be conditionally cooperative). This design is incentive-compatible because, after the match with other participants has been carried out, one randomly selected decision (i.e., unconditional or conditional) is used to compute the subjects' earnings. The screen eliciting conditional contributions is presented in Figure 1.

We ended up collecting laboratory data on 1,194 developers out of the pool of 2,534 eligible to participate. Right after the experiment and before payment, we asked subjects for some standard demographic information, that is, their age, gender, education, and salary range. These variables are described in Table 2.

We found that the population of OSS developers who answered our survey is generally young (32 years old on average) and overwhelmingly male (about 3% of developers report being female). The average developer in our experiment has a four-year college degree (BA, BS) with 17.5% of the population of developers having a lower qualification than a two-year college degree and almost half of the population (i.e., 49%) having a master's degree or a PhD. The average developer earns between $2,000 and $4,000 per month with 32% of the population earning less than $2,000 and 20% earning more than $7,500. These statics are consistent with survey studies on OSS developers (David and Shapiro 2008).

Figure 1. (Color online) The Decision Screen of the Conditional Public Goods Game



Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Table 2. Subject-Level Descriptive Statistics

Variable	Observations	Mean	Standard deviation	Minimum	Maximum
Age	1,192	32.20	8.42	16	72
Female	1,194	0.03	0.16	0	1
Education level	1,184	5.29	1.64	1	8
Income level	1,110	4.76	2.24	1	9

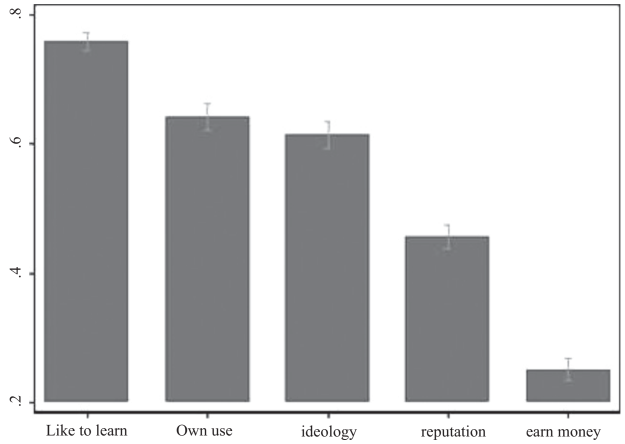
Notes. Age is measured in years. Education level: 1 = less than high school, 2 = high school, 3 = some college, 4 = two-year college degree, 5 = four-year college degree (BA, BS), 6 = master's degree, 7 = professional degree (MD, JD), 8 = doctoral degree. Income level (monthly): 1 = 0 USD, 2 = less than 1,000 USD; 3 = between 1,000 and 2,000 USD, 4 = between 2,000 and 3,000 USD, 5 = between 3,000 and 4,000 USD, 6 = between 4,000 and 5,000 USD, 7 = between 5,000 and 7,500 USD, 8 = between 7,500 and 10,000 USD, 9 = more than 10,000 USD.

Finally, we also asked developers a few questions about their motivations for contributing to OSS. Specifically, we followed the previous literature (see David and Shapiro 2008) and asked them to state their level of agreement with the following reasons for contributing to OSS on a scale ranging from 0 ("strongly disagree") to 10 ("strongly agree"): (i) to learn and develop new skills, (ii) to solve a problem that could not be solved by proprietary software, (iii) because I think software should not be a proprietary product, (iv) to build myself a reputation on the OSS developer scene, and (v) to make money. The means of those self-declared motivations are illustrated in Figure 2, together with their 95% confidence interval.

3.3. Field Data

In addition to the laboratory data, we collected panel data documenting team members' monthly code contributions to individual projects during the period March 2005–February 2013. We obtained this data from the Sourceforge Research Data Archive,[9] a project hosted at the University of Notre Dame that collected monthly data dumps from Sourceforge so as to make them available to the research community. Our panel data ends in February 2013 at which time Sourceforge put an end to its data-sharing agreement.

Figure 2. Developers' Reported Motives for Contributing to OSS



We collected this data for all team members belonging to (i) projects to which our starting set of 1,194 experimental subjects contributed and (ii) all the other projects on which their teammates (9,343 codevelopers) worked without their participation. We obtained a final sample of 5,557 OSS projects involving 10,537 developers. By the end of our time period, about half of those projects had failed and died (i.e., we observed no activity in those projects in the final 12 months of our time period).

For each developer, we collected monthly data on the number of code contributions (i.e., code commits) over those 96 consecutive months. A commit is a set of changes to the source code of a project that makes logical sense (i.e., implements a new feature or solves a bug). In addition, we extracted the creation date of each project and took advantage of the fact that OSS development teams often document the characteristics of their projects on the Sourceforge platform to collect additional project-level information. This includes license restrictiveness, popularity of the programming languages used, working languages used, and target user population (see Table 3).

Finally, we needed to address the challenge of reliably measuring the level of success of OSS projects. Given that OSS projects are made freely available for anyone to use, standard measures of popularity (e.g., sales) cannot be used as a proxy for success. In addition, because projects largely rely on voluntary contributions for their development, measures of input (e.g., code contributions) should be seen as an indicator of success in their own right. As a result, success needs to be defined at the project level as a function of both user popularity (i.e., "use") and community input (Crowston and Scozzi 2002, Crowston et al. 2004, Grewal et al. 2006, Van Antwerp and Madey 2010).

We achieved this goal by extracting Sourceforge's own ranking measure: the monthly "activity percentile" of each project, which combines the aforementioned dimensions to compute an exogenous, dynamic measure of success. The activity percentile is automatically calculated by Sourceforge and is prominently displayed on each project summary page. As Van Antwerp and Madey (2010, p. 2) put it, "projects with a high activity percentile are popular projects

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Table 3. Project-Level Descriptive Statistics

Variable	Observations	Mean	Standard deviation	Minimum	Maximum
Success score (activity percentile)	5,336	78.137	17.873	19.803	99.991
Project age (years)	5,558	5.36	2.74	0	43.10
License restrictiveness	4,871	2.35	0.75	1	3
Software aimed at end user	5,557	0.24	0.36	0	1
Team works in English	5,557	0.30	0.40	0	1
Mean popularity of programming languages	4,777	2.74	0.45	0	3.24

Notes. Project age is measured in years. License restrictiveness ranges from 1 (low restrictiveness) to 3 (high restrictiveness) as in Belenzon and Schankerman (2015). Software projects are considered to be aimed at end users when the team marks them as such (as opposed to being aimed at other developers or system administrators). We consider that a team works in English if English is listed as one of its working languages. Finally, to compute the mean popularity of the programming languages used at the project level, we take the log number of teams in our data set that report the use of any given programming language as a measure of overall programming language popularity. We then compute the average of these popularity scores at the project level (because many teams report using several programming languages at the same time).

since the measure is based on downloads, site views, development activity, and administrator activity."

Specifically, the measure aggregates (i) the size of the project user base, (ii) the intensity of contributors' development activity, and (iii) the use of project-related communication channels (see Online Appendix D for further details):

$$\begin{aligned} \textit{Activity Percentile} = {} & \frac{1}{3}\,\textit{User Traffic} \\ & + \frac{1}{3}\,\textit{Development Activity} \\ & + \frac{1}{3}\,\textit{Project Communication}. \end{aligned}$$

We end this section by summarizing our project-level variables in Table 3. The average activity percentile is relatively high in our project sample. This results from the fact that (i) our design tends to focus on a minority of collaboratively authored software projects within Sourceforge and (ii) the platform regularly purges its data from zombie projects. The average project in our data was about five years old by the end of our study. The oldest project in our data is 43 years old (i.e., older than Sourceforge itself) and was probably imported to Sourceforge from another hosting site. We see that projects generally tend to adopt relatively restrictive licenses (mean score of 2.4 out of 3). About a fourth of software projects are targeted at end users as opposed to other developers or system administrators, and 30% of teams use English as their working language. Finally, we counted the log number of teams in our data set that reported using any available programming language as a measure of its overall popularity in the community of developers. We then built a measure of the popularity of the programming languages used by each project by averaging those popularity scores at the project level.

4. Measuring Reciprocity

In this section, we describe the construction of our main explanatory variables, that is, our laboratory and field measures of developers' reciprocal preferences. As explained, lab-in-the-field experiments typically adopt the following methodology. At the time of the experiment, a sample of existing organizations is selected. Volunteers from these organizations participate in experimental games (the laboratory part), and the preferences elicited in those games are then related to measures of organizational performance (the field part). For organizations that are relatively stable, that is, less likely to die as a result of bad performance (e.g., a government agency), this is a powerful tool.

However, in the context of OSS as in many others (e.g., private firms operating in competitive markets or volunteer communities), organizations are much more volatile: some might grow, and others may quickly fail and die. Thus, when the experiment is run, only the participants in projects that have not yet died are available to be sampled. More generally, the activity level of a project and, consequently, the effective presence of its members on the research site may impact the likelihood that they participate in the experiment. This would then generate a sampling bias in which successful organizations are overrepresented compared with those that fail.

To overcome this problem, we propose another way to use our laboratory data in the field. The laboratory data serves to validate an analogous field measure of reciprocity based on archival data (or "activity traces"), which we can use in order to capture team members and projects that had already failed by the time of our study. We begin this section by describing our laboratory measure of reciprocity. We then discuss the construction of our related field measure before showing how both variables are correlated.

4.1. Experimental Measure of reciprocity

As described in Section 3.2.2, our subjects in the public goods game report both unconditional and conditional contribution decisions. As in the seminal paper from Fischbacher et al. (2001), we use the conditional

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

contribution decisions to compute a measure of reciprocal preferences at the developer level. We define this measure as the correlation between the player's conditional contribution decisions and the corresponding average contribution of the three other members of the player's group (from \$0 to \$10 as illustrated in Figure 1). This variable is distributed with a mean of 0.73 and a standard deviation of 0.45. Note that it can only capture positive, not negative, reciprocity because there is no opportunity in the standard public good games we used to pay a personal cost to decrease others' earnings. (The worst one can do to hurt others is to contribute zero, which is also the strategy that maximizes private payoffs.)

Although directly inspired by Fischbacher et al. (2001), the experimental measure of reciprocity that we define is computed as a simple correlation that captures the subject's willingness to be conditionally cooperative in the public goods game (i.e., behave reciprocally). This measure has the benefit of simplicity. It also has a direct analog in our field setting as we explain in the next section. By comparison, Fischbacher et al. (2001) classify their student subjects into three exclusive categories based on a visual examination of their conditional contribution patterns: (i) free-riders, who never contribute regardless of the contributions of others (this implies a correlation of zero in our setting); (ii) conditional cooperators, who match the contributions of the other members of their group (this implies a correlation of one in our setting); and (iii) conditional cooperators with a self-serving bias or weak conditional cooperators (Rustagi et al. 2010), who contribute to the public good, but generally less than others (this implies a positive correlation of less than one in our setting).[10]

4.2. Field Measure of Reciprocity

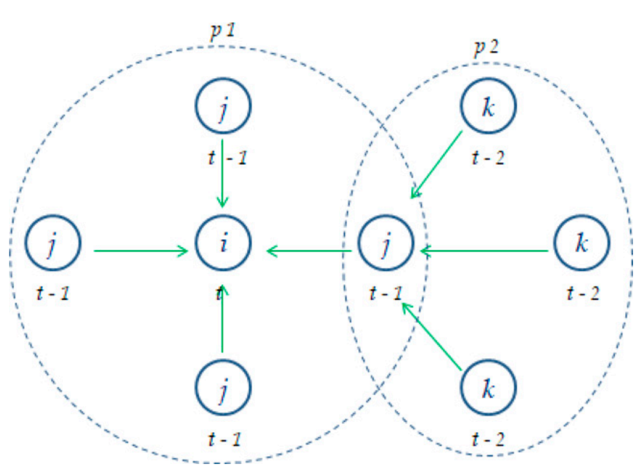
The next step in our study was to propose a field measure of reciprocity, inspired by the laboratory measure, that uses the observed patterns of contributions. The general idea was to measure the correlation between a participant's contributions in any given month with the sum of contributions made by the participant's team members in the previous month. This measure might, however, be biased because both the contributions of a given developer and those of the developer's team members in the previous period might be affected by common external factors. Suppose, for instance, that a productivity shock affects the project, such as one participant making a breakthrough that facilitates contributions by all the others. Such a shock, unobservable to us, could, in theory, simultaneously have affected the contribution level of a developer and those of the developer's team members in the previous period. We might, thus, incorrectly conclude, based on a positive correlation, that the individual was behaving reciprocally.

We, thus, proposed and built a field measure of reciprocity that corrected for this concern. For each developer, we computed the correlation between the developer's contributions at the project × month level and the *predicted* contributions of the developer's fellow team members in the previous month. This predicted measure used the variation in the sum of contributions made by the developer's own collaborators on the other projects that they pursued independently. By "independently," we mean that we required that the developer under consideration did not contribute to those other projects and, thus, never directly interacted with the collaborators of the developer's team members. We provide a formal description of the measure in Online Appendix C.

Figure 3 provides a graphical illustration of this strategy. For each developer i in our sample, we measure i's reactions in t to the monthly variation in contributions of the developer's team members j in $t-1$ as predicted by the exogenous variation in contributions of the developer's own team members k in $t-2$ on the projects that they pursue without developer i's involvement.

When computing the measure this way, we find that 3,700 out of the 8,250 developers for whom we can compute a field measure of reciprocal preferences in our final sample have a measure of reciprocity that is negative (i.e., they tend to decrease their level of contribution to a given project in the next period whenever their collaborators increase their own in the current period). This could possibly be an example of "altruistic" developers who care about providing as much public good as possible. Consequently, when their collaborators increase their contribution levels in a given team, they switch to contributing to other, relatively less well-developed open-source projects. Such preferences cannot be captured by our experimental

Figure 3. (Color online) Building a Measure of Reciprocity Based on Field Activity Traces



Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

design in which subjects are faced with a single common project. Alternatively, this pattern could be consistent with a scenario of substitutable inputs: if two programmers are perfect substitutes and one developer makes a contribution, this contribution can no longer be made by the other developer. Either way, in order to increase the conceptual link between our field and laboratory measures in which participants cannot contribute negatively to the public good, we define our field measure of reciprocity as the maximum of zero and of the correlation calculated earlier. This variable is distributed with a mean of 0.24, a standard deviation of 0.62, and min and max values of 0 and 11.05, respectively.

Table 4 explores the correlation between our laboratory and field measures of reciprocity. We can see from column (1) that both variables are strongly correlated: when the laboratory measure increases from zero to one, the field measure increases by an average of 0.5. Interestingly, the age, gender, and education level of the developers (column (2)) as well as their self-reported motives for contributing (column (3)), do not appear to be significantly related to their reciprocity preferences.

5. Reciprocity and the Success or Failure of Organizations

In the last part of the paper, we show that correcting for sampling bias generates new findings about the role of reciprocity in the success of organizations. We present these results in Section 5.1 and test the underlying theoretical mechanism in Section 5.2.

5.1. Reciprocity and Failure

We set the stage in Table 5, in which we link our experimental measure of reciprocity at the team level to average project-level success in our lab-in-the-field sample. In most cases, we only recruit one developer from each project through our experiment so that a single laboratory measure serves as a proxy for the average reciprocity level of the team. For 11% of the projects (i.e., 131 out of 1,143), we capture more than one developer per team and, therefore, average their laboratory reciprocity scores.

In all of the following analyses, we standardize the activity percentile variable. As a result, the coefficients reported represent the effect of reciprocity in terms of standard deviations of the success score in the underlying population of projects. When we apply the lab-in-the-field method directly (i.e., we do not correct for sampling bias), we obtain the same result as that found in the previous literature. In column (1), we see that moving from no to full reciprocity at the team level (i.e., the correlation between subjects' contributions in the experiment and the observed contributions of their team members shifts from zero to one) is associated with a significant 0.12 standard deviation increase in the success score.

In column (2), we add controls for project-level characteristics. Our coefficient of interest remains relatively unchanged (it slightly increases to a value of

Table 4. Correlation Between Laboratory and Field Measures

	(1) Reciprocity field	(2) Reciprocity field	(3) Reciprocity field
Reciprocity laboratory	0.50**	0.50**	0.49*
	(0.22)	(0.24)	(0.25)
Age		0.00	0.00
		(0.01)	(0.01)
Female		0.72	1.23
		(0.77)	(1.14)
Education level		−0.00	−0.02
		(0.06)	(0.07)
Motive: Ideology			0.25
			(0.22)
Motive: Like to learn			0.24
			(0.34)
Motive: Own use			0.07
			(0.20)
Motive: Establish reputation			0.33
			(0.31)
Motive: Pay			−0.03
			(0.19)
Regression type: cross-sectional, developer level			
R^2	0.01	0.03	0.06
Number of observations	231.00	230.00	210.00

Notes. Ordinary least squares estimates with robust standard errors in parentheses. *, **, *** denote statistical significance at the 10, 5 and 1% level, respectively. Column (1) includes no control, column (2) includes developers' self-reported demographics, and column (3) adds their reported motives for contributing to OSS.

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Table 5. Success and Survival: Laboratory-in-the-Field Sample

	(1) Success score	(2) Success score	(3) Success score
Mean reciprocity laboratory	0.12**	0.13**	0.16***
	(0.06)	(0.06)	(0.06)
Project age		0.00*	0.00*
		(0.00)	(0.00)
License restrictiveness		−0.01	−0.00
		(0.03)	(0.04)
Software aimed at end user		0.12*	0.15**
		(0.06)	(0.07)
Team works in English		0.19***	0.18***
		(0.06)	(0.06)
Project uses popular programming language		−0.09*	−0.04
		(0.05)	(0.05)
Mean age			0.01*
			(0.00)
Mean female			0.38***
			(0.12)
Mean education level			0.01
			(0.02)
Mean motive: Ideology			−0.03
			(0.08)
Mean motive: Like to learn			−0.11
			(0.13)
Mean motive: Own use			0.03
			(0.09)
Mean motive: Establish reputation			0.22**
			(0.09)
Mean motive: Pay			−0.00
			(0.05)
Regression type: cross-sectional, project level			
R^2	0.00	0.26	0.25
Number of observations	1,143.00	1,013.00	933.00

Notes. Ordinary least squares estimates with robust standard errors in parentheses. *, **, *** denote statistical significance at the 10, 5 and 1% level, respectively. Column (1) includes no control, column (2) includes project-level characteristics, and column (3) adds developers' self-reported demographics and motives for contributing to OSS (those scores are averaged if there is more than one subject per project).

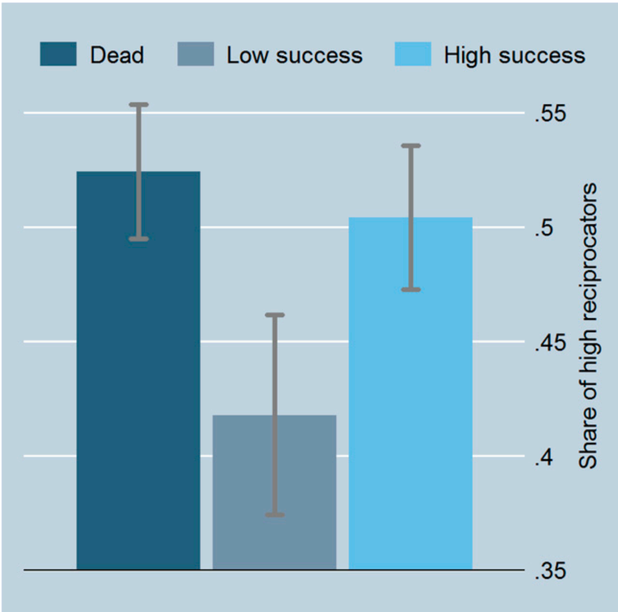
0.13). Among the control variables, using English as a working language is the one most significantly associated with success (i.e., 0.19 standard deviation increase in the success score). In column (3), we further add some controls for subjects' demographic characteristics and self-reported motives for contributing to OSS. This results in an increase in the statistical significance of our point estimate, which reaches a value of 0.16. In terms of magnitude, this represents 89% of the effect on success associated with the team mastering English as a working language (a significant advantage to attract contributions in a globalized virtual work environment). Finally, we also see from column (3) that teams that include more women and are relatively more motivated by their reputation generally achieve significantly higher success scores.

The rest of the paper shows that we obtain far more nuanced results if we use our field measure of reciprocity on our expanded sample of organizations and their members. We begin by providing a graphical representation of our main result in Figure 4. We divide projects into three categories: (i) dead projects, defined as those that did not receive any contribution in the last 12 months of our time period; (ii) active but low-success projects, which have an average activity percentile that is lower than the median in the sample; and (iii) active and high-success projects, which have an average activity percentile greater than the median in the sample.[11] For these three different types of projects, we plot the proportion of high reciprocators, defined as those that have an above-median field reciprocity measure.

Consistent with previous lab-in-the-field evidence, Figure 4 shows that low-success projects have an average of 40% of high reciprocators in their team, and high-success projects have 50% of high reciprocators: a 25% increase in proportion. Strikingly, however, dead projects do not significantly differ from highly successful ones in terms of the share of high reciprocators in their teams: 52% on average. Post hoc power analyses suggest that our study is properly powered to detect the empirical differences that we seek to uncover. Setting the probability of a type 1 error at $\alpha = 5\%$, the estimated power of a two-sided t-test of the difference in

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Figure 4. (Color online) Success and Failure of Projects with the Share of Conditional Cooperators



the proportion of high reciprocators between high and low success teams is 91.7%. Similarly, the estimated power of a test of the difference in proportion between dead and low-success teams is 98.5%.

This graphical analysis is confirmed in a regression framework. In Table 6, we examine the relationship between the proportion of high reciprocators in a given OSS team and project-level performance using our field measure of reciprocity. All regressions control for project-level characteristics (i.e., age, license restrictiveness, target user population, English as a working language, popularity of the programming languages used) and rely on robust standard errors for inference. The dependent variable in column (1) is a binary variable indicating whether the project is dead or not (i.e., did not receive any contribution in the last 12 months of our time period). Moving from an organization with no high reciprocators to one composed only of highly reciprocal members is associated with a 12% increase in the probability of project failure and death. Consistent with our results from Table 5, mastering English as a working language is the control variable most strongly associated with project failure. Specifically, teams that do not list English as one of their working languages are, on average, 8% more likely to fail. As a point of comparison, this represents 67% of the effect associated with working in a fully reciprocal team.

We next examine how the proportion of high reciprocators influences team success. To do so, we use the activity percentile variable. In column (2), we see that groups with a high share of highly reciprocal members achieve better success scores. Moving from a team with no high reciprocators to one composed only of this type is associated with a 0.17 standard deviation increase in the activity percentile. This relationship becomes stronger in column (3), in which we only retain projects that did not fail and remained active over the period.

In columns (4)–(6), we reproduce columns (1)–(3), but attempt to increase the precision of our estimates by excluding projects in which we cannot compute

Table 6. Success and Survival: Expanded Sample

	(1) Project dead	(2) Success score	(3) Success score	(4) Project dead	(5) Success score	(6) Success score
Share of high reciprocators	0.12***	0.17**	0.19**	0.15***	0.11	0.21**
	(0.04)	(0.08)	(0.09)	(0.05)	(0.08)	(0.10)
Project age	−0.00***	0.00***	0.00*	−0.00***	0.00***	0.00*
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
License restrictiveness	0.03*	−0.07*	−0.02	0.05**	−0.06	−0.01
	(0.02)	(0.03)	(0.04)	(0.02)	(0.04)	(0.04)
Software aimed at end user	0.05	0.06	0.09	0.05	0.07	0.09
	(0.04)	(0.06)	(0.06)	(0.04)	(0.07)	(0.07)
Team works in English	−0.08**	0.26***	0.20***	−0.08**	0.25***	0.18***
	(0.04)	(0.06)	(0.06)	(0.04)	(0.06)	(0.06)
Project uses popular programming language	−0.02	−0.02	−0.06	−0.02	−0.05	−0.06
	(0.04)	(0.05)	(0.05)	(0.04)	(0.05)	(0.05)
Regression type: cross-sectional, project level						
R^2	0.02	0.06	0.07	0.03	0.05	0.07
Number of observations	1,142.00	1,142.00	551.00	1,049.00	1,049.00	510.00

Notes. Ordinary least squares estimates with robust standard errors in parentheses. *, **, *** denote statistical significance at the 10, 5 and 1% level, respectively. The dependent variable in columns (1) and (4) takes the value one if the project died during our time period. The dependent variable in columns (2), (3), (5), and (6) is the project-level standardized success score. Columns (1) and (2) use our full sample of projects, and column (3) only retains projects that survive during our entire time period. Columns (4) and (5) exclude all projects from the full sample for which we cannot compute the field measure of reciprocity for at least one third of the team members (to increase the precision of our estimates), whereas column (6) restricts this sample to projects that survive during our entire time period.

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

our field measure of reciprocity for at least one third of the team members. With this restriction, the link between team-level reciprocity and the probability of failure becomes stronger (column (4)), and the link with project success becomes statistically insignificant (column (5)). However, we recover this significant positive relationship when we restrict our sample of projects to those that did not fail over the period (column (6)).

Our results at this point show that sampling bias may have prevented the extant lab-in-the-field literature from replicating laboratory-based results that imply that reciprocal preferences can also lead to increased failure rates at the team level. In Online Appendix E, we consider a number of robustness checks on these results. In particular, our results are robust to various ways of defining project death and field reciprocity. We can also exclude the interpretation according to which dead projects would actually be the most successful ones—having reached a mature stage at which contributions are no longer needed—by showing that (i) projects are more likely to fail at an early development stage and (ii) mature projects are actually the ones that are most actively developed (see Online Appendix E). One last concern could be particularly worrisome, so we address it in the main text. Namely, our field measure of reciprocity could capture the effect of individual-level characteristics other than reciprocal preferences, such as developers' cognitive abilities.

To demonstrate that omitted variable bias is unlikely to influence our field results, Table 7 reproduces the first three columns of Table 6. However, it considers the sample of projects that only have one contributor, that is, for which reciprocity preferences should not normally determine their success or failure. When we do this, we fail to find any strong statistical link between each developer's level of reciprocity and either the success of the developer's project or the probability that it eventually fails and dies. This suggests that our results are indeed motivated by the reciprocal dynamics that we seek to capture at the team level.

5.2. Mechanism: The Reciprocity Roller Coaster

In the laboratory, reciprocal preferences typically work as a catalyst, which equally amplifies positive and negative contribution dynamics at the group level. As a result, the presence of many highly reciprocal members in a team leads to the decay of cooperation whenever some observe that others do not match their contribution level in any given period (Fischbacher and Gächter 2010). At the same time, strong reciprocity makes it possible to sustain high levels of cooperation when everybody contributes at a relatively high rate (Page et al. 2005).

Our paper is the first to provide a field test of this microlevel mechanism. We hypothesize that highly reciprocal individuals react positively to increased contributions from team members but also decrease their contributions at a higher rate if they believe that others do not exert sufficient effort. An organization with a high proportion of strong reciprocators is, therefore, likely to perform at above-average levels when contributions are already relatively high. On the other hand, the same organization is predicted to have a harder time recovering from a period of inactivity, resulting in a significant increase in the probability of death. Note that, for cooperation to collapse within this framework, it is sufficient for team

Table 7. Projects with a Single Contributor

	(1) Project dead	(2) Success score	(3) Success score
High reciprocator	0.01	0.10*	−0.06
	(0.02)	(0.05)	(0.08)
Project age	0.00***	−0.00***	−0.00**
	(0.00)	(0.00)	(0.00)
License restrictiveness	0.01	−0.06	−0.13**
	(0.01)	(0.04)	(0.06)
Software aimed at end user	0.01	0.28***	0.26**
	(0.03)	(0.08)	(0.13)
Developer works in English	−0.11***	0.35***	0.36***
	(0.03)	(0.08)	(0.14)
Project uses popular programming language	0.02	0.02	0.05
	(0.02)	(0.06)	(0.12)
Regression type: cross-sectional, project level			
R^2	0.04	0.04	0.08
Number of observations	1,433.00	1,433.00	265.00

Notes. Ordinary least squares estimates with robust standard errors in parentheses. *, **, *** denote statistical significance at the 10, 5 and 1% level, respectively. This table replicates columns (1)–(3) of Table 6 using the sample of single-authored projects. The dependent variable in column (1) takes the value one if the project died during our time period. The dependent variable in columns (2) and (3) is the project-level standardized success score. Columns (1) and (2) use our full sample of projects, and column (3) only retains projects that survive during our entire time period.

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

members to *believe* that others are not contributing. This is especially important in field contexts in which actual contributions (i.e., contributions that are observable to other team members) often imperfectly reflect individuals' effort levels (i.e., their willingness to cooperate). Over the course of many months or years, some periods of inactivity in our virtual team setting doubtlessly result from idiosyncratic shocks at the individual or project level. Such shocks could then affect beliefs about individual effort levels and ultimately determine the dynamics of cooperation within the team.

In Table 8, we take advantage of the panel structure of our data on individual team contributions to analyze the dynamics of cooperation within projects at a monthly frequency over the eight-year period covered by our study. To do this, we run developer-level panel regressions in which our unit of observation is at the developer × project × month level.

We first analyze whether the share of highly reciprocal developers at the team level impacts the probability that a project recovers from a period of inactivity (column (1)). In this regression, we define inactivity as a period of three consecutive months without any team contribution and include an interaction term indicating whether the developer is of the high reciprocity type. Our dependent variable is a dummy indicating whether the developer made any contribution to the project in the following month. The regression controls for all available project-level characteristics and reports robust standard errors clustered at the project level. We see that facing a period of inactivity increases the probability that the developer makes no contribution in the subsequent month by 13% when the developer is not highly reciprocal. Consistent with our hypothesis, this probability increases by an additional 10% in the case of a highly reciprocal developer. Conversely, when contributions have been made to the project in previous periods, highly reciprocal developers have a 10% lower probability of making no contribution in the following period.

These results are confirmed in column (2), in which we take the total number of code contributions (or commits) made by a developer to a project in any given month as an alternative dependent variable. The specification is the same as in column (1) except that we now include developer-level fixed effects in order to properly account for unobservable characteristics at the individual level. We obtain similar results. Facing

Table 8. Microlevel Mechanism

	(1) No contribution	(2) Total no. of commits	(3) Total no. of commits
No contribution over last three periods	0.13***	−1.25**	
	(0.01)	(0.61)	
Interaction with high reciprocity type	0.10***	−2.28***	
	(0.01)	(0.79)	
Above-median contributions over last three periods			2.31**
			(1.03)
Interaction with high reciprocity type			3.16**
			(1.33)
High reciprocity type	−0.10***	x	x
	(0.01)		
Project age	0.00***	−0.00**	−0.00**
	(0.00)	(0.00)	(0.00)
License restrictiveness	0.00	0.08	0.08
	(0.00)	(0.10)	(0.10)
Software aimed at end user	−0.01	0.07	0.06
	(0.00)	(0.18)	(0.18)
Team works in English	−0.01	0.44*	0.50*
	(0.00)	(0.25)	(0.27)
Project uses popular programming language	0.00	−0.31*	−0.34*
	(0.00)	(0.17)	(0.19)
Regression type: panel, developer × project × month level			
Developer fixed effect	No	Yes	Yes
R^2	0.12	0.01	0.02
Number of observations	2.1e+05	2.1e+05	2.1e+05

Notes. Ordinary least squares estimates with robust standard errors clustered at the developer level in parentheses. *, **, *** denote statistical significance at the 10, 5 and 1% level, respectively. The dependent variable in column (1) is a dummy variable indicating whether no contribution is made by the developer in a given project × month. We regress it on a variable that captures whether no contributions were made by team members in the previous three months, interacted with the social type of the developer (highly reciprocal or not). The dependent variable in columns (2) and (3) is the number of code contributions (or commits) made by the developer in a given project × month. We regress it on a variable that captures whether team members made above-median contributions to the project in the previous three months, interacted with the social type of the developer (highly reciprocal or not). Note that columns (2) and (3) include developer-level fixed effects (this is why the coefficient on the variable "high reciprocity type" is dropped from those regressions).

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

a period of inactivity decreases the average number of contributions made in the current period by 1.25 for nonreciprocators. In the case of high-reciprocity types, this number decreases to −3.54, a threefold increase in magnitude.[12]

Finally, column (3) relies on the same econometric specification as column (2) but focuses on the benefits of reciprocal preferences at the team level, that is, the fact that they typically reinforce cooperative dynamics. For each project, we define a period of "high activity" as a period in which the number of contributions made in the three previous months was higher than the median number of monthly contributions over the history of the project. We find that non-highly reciprocal developers make an average of 2.3 more contributions following a period of high activity, and highly reciprocal ones make an average of 5.5 more contributions: a 150% increase in contribution levels. Taken together, these results provide evidence for the microlevel mechanism that we posit behind our aggregate results: highly reciprocal organizations are both more highly represented among top performers (at least conditional on survival) but are also more likely to experience cooperation breakdowns and fail.

6. Conclusion

This paper provides the first field evidence that strong reciprocity is a double-edged sword for organizations. We study the context of OSS, in which team members need to cooperate voluntarily toward the provision of a public good, and link laboratory and field measures of reciprocity to project-level archival data to demonstrate that highly reciprocal teams are not necessarily more successful. They outperform other teams conditional on survival but are also more likely to experience cooperation breakdowns. Moreover, we use the detailed panel structure of our data to pinpoint the microlevel mechanism behind these aggregate results. Reciprocal preferences work as a catalyst at the team level: they reinforce the cooperative equilibrium as long as team members receive positive signals, leading to top-notch performance, but they also increase the probability of a cooperation breakdown in the face of negative news.

At a methodological level, our paper adds to the fast-growing lab-in-the-field literature. This methodology is a powerful tool for organizational research. The experimenter relies on validated laboratory paradigms to elicit psychological traits or preferences that are difficult to identify "in the wild" and links them to outcomes of theoretical interest in the field. In this process, the experimenter can maintain a close connection with laboratory-based research, thus alleviating the tension between internal and external validity in experimental research (Gneezy and Imas 2017). At the same time, the lab-in-the-field method is most useful to disentangle competing microlevel mechanisms that could potentially drive aggregate empirical regularities, therefore contributing to the microfoundation movement in organizational behavior (Felin and Foss 2005, Felin et al. 2015, Bitektine et al. 2018).

Several previous studies use this methodology to test the link between reciprocity and group-level performance (Barr and Serneels 2009, Rustagi et al. 2010, Carpenter and Seki 2011). Surprisingly, all of them report an unambiguous positive relationship between reciprocal preferences and organizational success. This result represents a relative disconnect with the laboratory literature, in which reciprocal preferences trigger the decay of cooperation whenever highly reciprocal types observe that others' efforts are not commensurate with their own. We hypothesize that this disconnect may result from the fact that the lab-in-the-field method requires researchers to run their experiments directly within the natural environment of interest. We argue that this can introduce a significant sampling bias: previously failed organizations are much less likely to be included in the design because former members may no longer be present on the research site. Thus, because they rely on a design that tends to ignore failed projects and their members, previous studies may not capture the dark side of reciprocal dynamics at the organizational level. To address this problem, we propose an additional way to use laboratory data in the field, that is, to validate an analogous field measure based on archival data (or field activity traces) to expand our research sample, notably in the direction of previously failed organizations.

Our field results suggest that mechanisms that screen workers according to their social type are unlikely to generate a self-sustained cooperative equilibrium. In an uncertain world in which organizations are subject to adverse shocks and imperfect monitoring, implicit norms define the nature of the cooperation game that agents play on a daily basis, and relational contracts guide the interpretation of the (noisy) signals received by individual members (Chassang 2010, Gibbons and Henderson 2012, Acemoglu and Jackson 2015). As with all others, highly reciprocating teams are subject to "unforeseen contingencies" (Kreps et al. 1996). Depending on the members' previous shared history (or relational knowledge), such adverse shocks may or may not trigger the unraveling of cooperation within the organization. Future experimental research on cooperation should seek to explore how relational contracts can be defined in a way that minimizes the probability of inefficient cooperation breakdowns among reciprocally minded workers (Fudenberg et al. 2012, Gibbons et al. 2020).

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Acknowledgments

The authors gratefully acknowledge logistical support from the Sciences Po médialab. The authors are indebted to Sourceforge.net for endorsing this study and to the Sourceforge Research Data Archive at the University of Notre Dame for granting privileged access to their data. The authors are grateful to Anne l'Hôte, Romain Guillebert, and Matt van Antwerp for their outstanding research assistance. Final thanks go to the open-source software developers who took part in this study. The authors have reported all measures, conditions, data exclusions, and tests performed within the context of this paper. The code and documentation for the online public goods game used in this paper can be retrieved at https://github.com/medialab/trustlab. The raw data and code on which this paper is based can be found at https://doi.org/10.21410/7E4/UCH21T. The authors have no conflicts of interest to disclose.

Endnotes

[1] By "organization," we mean any group or team of individuals who work together toward some shared purpose. This could be a nation, a firm, a not-for-profit organization, or an open-innovation community (e.g., a research team).

[2] Among many other examples, see Gittell (2000) and Gittell et al. (2004) for the U.S. airlines industry, Ichniowski et al. (1997) for the steel industry, Dyer and Nobeoka (2000) and Helper and Henderson (2014) for the automobile industry, and Gittell et al. (2010) for the healthcare sector.

[3] A few exceptions include Bereby-Meyer and Roth (2006), Kunreuther et al. (2009), Xiao and Kunreuther (2016), Grechenig et al. (2010), and Ambrus and Greiner (2012). In general, those experimental papers show that sustaining a cooperative equilibrium is more difficult in uncertain environments.

[4] There is a similar literature in political science: Finan and Schechter (2012) experimentally elicit reciprocal preferences in a population of community leaders in Paraguay and show that highly reciprocal village chiefs are more likely to be targeted by politicians for vote-buying purposes. Similarly, Baldassarri and Grossman (2011) and Grossman and Baldassarri (2012) demonstrate that cooperation in a repeated public goods game in which a "leader" has the ability to punish group members based on past contributions predicts field cooperation among Ugandan farmers but only when the leader is elected by the people, which corresponds to the way chiefs are appointed in this field setting. Gilligan et al. (2014) use exogenous variation in the extent to which local communities in Nepal were affected by civil war to show that stronger exposure to violence can lead to collective coping through social cohesion as measured by subjects' behavior in a standard public goods game. In a related paper, Blair (2018) runs lab-in-the-field experiments in Liberia to show that exposure to war-time violence increases the government's ability to instruct citizens to make voluntary contributions to public goods.

[5] The formal definition of reciprocity is strikingly similarly in economics (see Dufwenberg and Kirchsteiger 2004, Sobel 2005, Falk and Fischbacher 2006). Furthermore, the result that individuals differ in how strongly they endorse the norm of reciprocity, obtained in the context of the experimental literature on the decay of cooperation in repeated public goods experiments (Fischbacher et al. 2001, Fischbacher and Gächter 2010), is previously established in the context of laboratory-based tests of SET (see Eisenberger et al. 1987).

[6] Note that the possibility of decentralized punishment does not necessarily lead to enhanced cooperation and welfare, especially in noisy environments such as those considered in this paper (see Grechenig et al. 2010, Ambrus and Greiner 2012). See also the designs in which punished subjects decide to engage in welfare reducing "antisocial punishment" (Denant-Boemont et al. 2007, Herrmann et al. 2008, Nikiforakis 2008, Nikiforakis et al. 2012).

[7] Because of this, relational contracts are also very difficult to imitate and can represent a significant source of competitive advantage. See Gibbons and Henderson (2012) for a discussion of relational contracts at Lincoln Electric, Toyota, and Merck as well as Helper and Henderson (2014) for General Motors.

[8] Two main features define the restrictiveness of a project license: (i) the extent to which the code and any of its modifications can subsequently be embedded in commercial software and (ii) whether modifications to the code have to remain open source (i.e., free to use, study, share, and modify by anyone).

[9] See http://Archive, www3.nd.edu/oss/Data/data.html and Van Antwerp and Madey (2008).

[10] Some differences between our pool of OSS developers and the populations of students typically used in laboratory experiments are noteworthy. Only 4% of our subjects could be classified as free-riders (compared with 20%–30% in student populations), 48% were perfect reciprocators, and 41% were weak reciprocators. Finally, 7% of our subjects unconditionally contributed *all* of their endowment to the public good (an altruistic pattern of contributions that is typically not observed among students).

[11] For the purpose of this figure, we only include projects with at least two developers and for which we can compute a field measure of reciprocity for at least a third of the team members. Our subsequent regression analysis releases these constraints to establish the robustness of this result.

[12] In order to compute the effect size on highly reciprocal types, the baseline coefficient is added to the interaction term: $-1.25 -2.28 = -3.54$.

References

Acemoglu D, Jackson MO (2015) History, expectations, and leadership in the evolution of social norms. *Rev. Econom. Stud.* 82(2): 423–456.

Ambrus A, Greiner B (2012) Imperfect public monitoring with costly punishment: An experimental study. *Amer. Econom. Rev.* 102(7): 3317–3332.

Anthony D (2005) Cooperation in microcredit borrowing groups: Identity, sanctions, and reciprocity in the production of collective goods. *Amer. Sociol. Rev.* 70(3):496–515.

Baldassarri D, Grossman G (2011) Centralized sanctioning and legitimate authority promote cooperation in humans. *Proc. Natl. Acad. Sci. USA* 108(27):11023–11027.

Barr A, Serneels P (2009) Reciprocity in the workplace. *Experiment. Econom.* 12(1):99–112.

Bartling B, Fehr E, Schmidt KM (2012) Screening, competition, and job design: Economic origins of good jobs. *Amer. Econom. Rev.* 102(2):834–864.

Belenzon S, Schankerman M (2015) Motivation and sorting of human capital in open innovation. *Strategic Management J.* 36(6):795–820.

Benkler Y (2002) Coase's penguin, or, linux and "the nature of the firm." *Yale Law J.* 112(3):369–446.

Bereby-Meyer Y, Roth AE (2006) The speed of learning in noisy games: Partial reinforcement and the sustainability of cooperation. *Amer. Econom. Rev.* 96(4):1029–1042.

Bitektine A, Lucas J, Schilke O (2018) Institutions under a microscope: Experimental methods in institutional theory. Bryman A, Buchanan DA, eds. *Unconventional Methodology in Organization and Management Research* (Oxford University Press, Oxford, UK), 147–167.

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

Blair R (2018). Legitimacy after violence: Evidence from two laboratory-in-the-field experiments in Liberia. Preprint, submitted January 23, https://dx.doi.org/10.2139/ssrn.2326671.

Blau PM (1964) *Exchange and Power in Social Life* (Transaction Publishers).

Carpenter J, Seki E (2011) Do social preferences increase productivity? Field experimental evidence from fishermen in Toyama Bay. *Econom. Inquiry* 49(2):612–630.

Charness G, Villeval M-C (2009) Cooperation and competition in intergenerational experiments in the field and the laboratory. *Amer. Econom. Rev.* 99(3):956–978.

Charness G, Yang C-L (2014) Starting small toward voluntary formation of efficient large groups in public goods provision. *J. Econom. Behav. Organ.* 102:119–132.

Chassang S (2010) Building routines: Learning, cooperation, and the dynamics of incomplete relational contracts. *Amer. Econom. Rev.* 100(1):448–465.

Chaudhuri A (2011) Sustaining cooperation in laboratory public goods experiments: A selective survey of the literature. *Experiment. Econom.* 14(1):47–83.

Cinyabuguma M, Page T, Putterman L (2005) Cooperation under the threat of expulsion in a public goods experiment. *J. Public Econom.* 89(8):1421–1435.

Cook KS, Emerson RM (1978) Power, equity and commitment in exchange networks. *Amer. Sociol. Rev.* 43(5):721–739.

Cook KS, Cheshire C, Rice ER, Nakagawa S (2013) Social exchange theory. *Handbook of Social Psychology. Handbooks of Sociology and Social Research.* (Springer Dordrecht), 61–88.

Cropanzano R, Mitchell MS (2005) Social exchange theory: An interdisciplinary review. *J. Management* 31(6):874–900.

Cropanzano R, Anthony EL, Daniels SR, Hall AV (2017) Social exchange theory: A critical review with theoretical remedies. *Acad. Management Ann.* 11(1):479–516.

Crowston K, Scozzi B (2002) Open source software projects as virtual organisations: Competency rallying for software development. *IEE Proc. Software* 149(1):3–17.

Crowston K, Annabi H, Howison J, Masango C (2004) Toward a portfolio of floss project success measures. In Workshop on Open Source Software Engineering. *26th Internat. Conf Software Engrg.* (IET, Edinburgh), 29–33.

Crowston K, Wei K, Howison J, Wiggins A (2012) Free/libre open-source software development: What we know and what we do not know. *ACM Comput. Surveys* 44(2):71–35.

Dahlander L, Magnusson MG (2005) Relationships between open source software companies and communities: Observations from Nordic firms. *Res. Policy* 34(4):481–493.

David PA, Shapiro JS (2008) Community-based production of open-source software: What do we know about the developers who participate? *Inform. Econom. Policy* 20(4):364–398.

Denant-Boemont L, Masclet D, Noussair CN (2007) Punishment, counterpunishment and sanction enforcement in a social dilemma experiment. *Econom. Theory* 33(1):145–167.

Dufwenberg M, Kirchsteiger G (2004) A theory of sequential reciprocity. *Games Econom. Behav.* 47(2):268–298.

Dyer JH, Nobeoka K (2000) Creating and managing a high-performance knowledge-sharing network: The Toyota case. *Strategic Management J.* 21(3):345–367.

Eisenberger R, Cotterell N, Marvel J (1987) Reciprocation ideology. *J. Personality Soc. Psych.* 53(4):743–750.

Emerson RM (1976) Social exchange theory. *Annual Rev. Sociol.* 2(1): 335–362.

Falk A, Fischbacher U (2006) A theory of reciprocity. *Games Econom. Behav.* 54(2):293–315.

Faraj S, Jarvenpaa SL, Majchrzak A (2011) Knowledge collaboration in online communities. *Organ. Sci.* 22(5):1224–1239.

Fehr E, Gächter S (2000a) Cooperation and punishment in public goods experiments. *Amer. Econom. Rev.* 90(4):980–994.

Fehr E, Gächter S (2000b) Fairness and retaliation: The economics of reciprocity. *J. Econom. Perspect.* 14(3):159–181.

Fehr E, Leibbrandt A (2011) A field study on cooperativeness and impatience in the tragedy of the commons. *J. Public Econom.* 95(9–10):1144–1155.

Felin T, Foss NJ (2005) Strategic organization: A field in search of micro-foundations. *Strategic Organ.* 3(4):441–455.

Felin T, Foss NJ, Ployhart RE (2015) The microfoundations movement in strategy and organization theory. *Acad. Management Ann.* 9(1):575–632.

Finan F, Schechter L (2012) Vote-buying and reciprocity. *Econometrica* 80(2):863–881.

Fischbacher U, Gächter S (2010) Social preferences, beliefs, and the dynamics of free riding in public goods experiments. *Amer. Econom. Rev.* 100(1):541–556.

Fischbacher U, Gächter S, Fehr E (2001) Are people conditionally cooperative? Evidence from a public goods experiment. *Econom. Lett.* 71(3):397–404.

Fudenberg D, Rand DG, Dreber A (2012) Slow to anger and fast to forgive: Cooperation in an uncertain world. *Amer. Econom. Rev.* 102(2):720–749.

Gächter S, Renner E, Sefton M (2008) The long-run benefits of punishment. *Sci.* 322(5907):1510–1510.

Ghosh RA (2007) Economic impact of open source software on innovation and the competitiveness of the information and communication technologies (ICT) sector in the EU.

Gibbons R, Henderson R (2012) Relational contracts and organizational capabilities. *Organ. Sci.* 23(5):1350–1364.

Gibbons RS, Grieder M, Herz H, Zehnder C (2020) Building an equilibrium: Rules vs. principles in relational contracts. CESifo Working Paper No. 7871, Munich, Germany.

Gilligan MJ, Pasquale BJ, Samii C (2014) Civil war and social cohesion: Laboratory-in-the-field evidence from Nepal. *Amer. J. Political Sci.* 58(3):604–619.

Gittell JH (2000) Organizing work to support relational co-ordination. *Internat. J. Human Resource Management* 11(3):517–539.

Gittell JH, Seidner R, Wimbush J (2010) A relational model of how high-performance work systems work. *Organ. Sci.* 21(2):490–506.

Gittell JH, von Nordenflycht A, Kochan TA (2004) Mutual gains or zero sum? Labor relations and firm performance in the airline industry. *Indust. Labor Relations Rev.* 57(2):163–180.

Gneezy U, Imas A (2017) Laboratory in the field: Measuring preferences in the wild. Banerjee AV, Duflo E, eds. *Handbook of Economic Field Experiments*, vol. 1 (Elsevier), 439–464.

Gouldner AW (1960) The norm of reciprocity: A preliminary statement. *Amer. Sociol. Rev.* 25(2):161–178.

Grechenig K, Nicklisch A, Thöni C (2010) Punishment despite reasonable doubt—A public goods experiment with sanctions under uncertainty. *J. Empirical Legal Stud.* 7(4):847–867.

Greenstein S, Nagle F (2014) Digital dark matter and the economic contribution of Apache. *Res. Policy* 43(4):623–631.

Grewal R, Lilien GL, Mallapragada G (2006) Location, location, location: How network embeddedness affects project success in open source systems. *Management Sci.* 52(7):1043–1056.

Grossman G, Baldassarri D (2012) The impact of elections on cooperation: Evidence from a lab-in-the-field experiment in Uganda. *Amer. J. Political Sci.* 56(4):964–985.

Helper S, Henderson R (2014) Management practices, relational contracts, and the decline of General Motors. *J. Econom. Perspect.* 28(1):49–72.

Hergueux J, Jacquemet N (2015) Social preferences in the online laboratory: A randomized experiment. *Experiment. Econom.* 18(2): 251–283.

Herrmann B, Thöni C, Gächter S (2008) Antisocial punishment across societies. *Science* 319(5868):1362–1367.

Hertel G, Niedner S, Herrmann S (2003) Motivation of software developers in open source projects: An internet-based

Downloaded from informs.org by [114.214.215.95] on 15 February 2023, at 00:22 . For personal use only, all rights reserved.

survey of contributors to the Linux kernel. *Res. Policy* 32(7): 1159–1177.

Homans GC (1961) *Social Behavior: Its Elementary Forms* (Harcourt, Brace, San Diego).

Ichniowski C, Shaw K, Prennushi G (1997) The effects of human resource management practices on productivity: A study of steel finishing lines. *Amer. Econom. Rev.* 87(3):291–313.

Karlan DS (2005) Using experimental economics to measure social capital and predict financial decisions. *Amer. Econom. Rev.* 95(5): 1688–1699.

Kellogg KC (2009) Operating room: Relational spaces and microinstitutional change in surgery. *Amer. J. Sociol.* 115(3):657–711.

Kosfeld M, Rustagi D (2015) Leader punishment and cooperation in groups: Experimental field evidence from commons management in Ethiopia. *Amer. Econom. Rev.* 105(2):747–783.

Kosfeld M, von Siemens FA (2011) Competition, cooperation, and corporate culture. *RAND J. Econom.* 42(1):23–43.

Kreps DM, Alt JE, Shepsle KA (1996) Corporate culture and economic theory. *Firms, Organizations and Contracts* (Oxford University Press, Oxford, UK), 221–275.

Kunreuther H, Silvasi G, Bradlow E, Small D (2009) Bayesian analysis of deterministic and stochastic prisoner's dilemma games. *Judgment Decision Making* 4(5):363–384.

Lakhani KR, Wolf RG (2005) Why hackers do what they do: Understanding motivation and effort in free/open source software projects. Feller J, Fitzgerald B, Hissam S, Lakhani KR, eds. *Perspectives on Free and Open Source Software* (MIT Press, Cambridge, MA), 3–22.

Lazer D, Pentland A, Adamic L, Aral S, Barabasi A-L, Brewer D, Christakis N, et al. (2009) Computational social science. *Sci.* 323(5915):721–723.

Ledyard JO (1995) Public goods: A survey of experimental research. Kagel JH, Roth AE, eds. *The Handbook of Experimental Economics* (Princeton University Press, Princeton), 111–194.

Lee GK, Cole RE (2003) From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organ. Sci.* 14(6):633–649.

Leibbrandt A (2012) Are social preferences related to market performance? *Experiment. Econom.* 15(4):589–603.

Levine SS, Prietula MJ (2013) Open collaboration for innovation: Principles and performance. *Organ. Sci.* 25(5):1414–1433.

Marlow J, Dabbish L, Herbsleb J (2013) Impression formation in online peer production: activity traces and personal profiles in Github. *Proc. 2013 Conf. Comput. Supported Cooperative Work* (ACM, New York), 117–128.

Maurer SM, Scotchmer S (2006) Open source software: The new intellectual property paradigm. Technical report, National Bureau of Economic Research, Cambridge, MA.

Mayer KJ, Argyres NS (2004) Learning to contract: Evidence from the personal computer industry. *Organ. Sci.* 15(4):394–410.

Molm LD (2003) Theoretical comparisons of forms of exchange. *Sociol. Theory* 21(1):1–17.

Molm LD, Takahashi N, Peterson G (2000) Risk and trust in social exchange: An experimental test of a classical proposition. *Amer. J. Sociol.* 105(5):1396–1427.

Montgomery JD (1996) The structure of social exchange networks: A game-theoretic reformulation of Blau's model. *Sociol. Methodology* 26:193–225.

Nikiforakis N (2008) Punishment and counter-punishment in public good games: Can we really govern ourselves? *J. Public Econom.* 92(1–2):91–112.

Nikiforakis N, Noussair CN, Wilkening T (2012) Normative conflict and feuds: The limits of self-enforcement. *J. Public Econom.* 96(9–10):797–807.

O'Mahony S (2003) Guarding the commons: How community managed software projects protect their work. *Res. Policy* 32(7): 1179–1198.

Page T, Putterman L, Unel B (2005) Voluntary association in public goods experiments: Reciprocity, mimicry and efficiency. *Econom. J. (London)* 115(506):1032–1053.

Roth AE (1995) Introduction to experimental economics. Kagel JH, Roth AE, eds. *The Handbook of Experimental Economics* (Princeton University Press, Princeton, NJ), 3–109.

Rustagi D, Engel S, Kosfeld M (2010) Conditional cooperation and costly monitoring explain success in forest commons management. *Science* 330(6006):961–965.

Scacchi W (2007) Free/open source software development: Recent research results and emerging opportunities. *Sixth Joint Meeting Eur. Software Engrg. Conf. ACM SIGSOFT Sympos. Foundations Software Engrg. Companion Papers* (ACM, New York), 459–468.

Schram A (2005) Artificiality: The tension between internal and external validity in economic experiments. *J. Econom. Methodology* 12(2):225–237.

Sobel J (2005) Interdependent preferences and reciprocity. *J. Econom. Literature* 43(2):392–436.

Van Antwerp M, Madey G (2008) Advances in the Sourceforge research data archive (SRDA). *Fourth Internat. Conf. Open Source Systems.*

Van Antwerp M, Madey G (2010) The importance of social network structure in the open source software developer community. *43rd Hawaii Internat. Conf. System Sci.* (IEEE), 1–10.

von Hippel E, von Krogh G (2003) Open source software and the "private-collective" innovation model: Issues for organization science. *Organ. Sci.* 14(2):209–223.

von Krogh G, von Hippel E (2006) The promise of research on open source software. *Management Sci.* 52(7):975–983.

Walli S, Gynn D, von Rotz B (2005) The growth of open source software in organizations. Technical report, Optaros Consulting, Boston.

Weber S (2004) *The Success of Open Source* (Harvard University Press, Cambridge, MA).

Xiao E, Kunreuther H (2016) Punishment and cooperation in stochastic social dilemmas. *J. Conflict Resolution* 60(4):670–693.

Jérôme Hergueux is an assistant research professor at the French National Center for Scientific Research (CNRS), a research affiliate at the Center for Law and Economics at ETH Zurich, and a faculty associate at the Berkman Klein Center for Internet & Society at Harvard University. He is a behavioral economist operating at the boundaries between psychology, economics and computational social science. His research focuses on online communities, social computing, and digitization

Emeric Henry is a professor of economics at Sciences Po. He is a microeconomist using theory, experimental and empirical methods to study questions in law and economics.

Yochai Benkler is the Berkman Professor of Entrepreneurial Legal Studies at Harvard Law School and faculty codirector of the Berkman Klein Center for Internet and Society at Harvard University. Since the 1990s, he has played a role in characterizing the role of information commons and decentralized collaboration to innovation, information production, and freedom in the networked economy and society.

Yann Algan is dean of the School of Public Affairs and professor of economics at Sciences Po. His research focuses on populism; social capital; well-being; the evaluation of the public policies, particularly in the area of education and employment; and the digital transformation of public action. He is a member of the *Institut Universitaire de France*, senior editor of the review *Economic Policy*, and recipient of two ERC projects from the European Commission.