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

March 5, 2020

Abstract

“Lab-in-the-field” experiments unambiguously point to reciprocity as an important driver of the success of real-world organizations. This empirical result is (partly) at odds with laboratory research on the private provision of public goods, where reciprocal preferences can cause a breakdown in cooperation. We argue that the lab-in-the-field methodology is a powerful tool for organizational research, but that it might also suffer from sampling bias: researchers collect data from existing organizations, i.e., those that did not fail and disappear. Using the context of open source software virtual teams – where the full history of both failed and surviving projects can be recovered – we propose to address this issue by leveraging lab data both directly to predict outcomes and to validate a generalizable measure of field reciprocity, which can be computed for both active and failed projects. Using this alternative approach to lab-in-the-field, we show that even though virtual teams with a larger share of reciprocators are more successful when they survive, they are also more likely to fail. We leverage the panel structure of our data to show that reciprocal preferences work as a catalyst: they reinforce team dynamics and accelerate success during productive times, but also make it harder to recover from periods of inactivity.

Keywords: Cooperation; Reciprocity; Social Exchange; Organizational Behavior; Virtual Teams; Open Source Software.

JEL Classification: M54; M21; H41.
1 Introduction

Public good problems – that is, instances of social dilemma where the interest of a group is not aligned with that of its individual members – are ubiquitous within organizations. Individual group members often need to cooperate voluntarily (that is, act in the interest of their group at a cost to themselves) in order to achieve efficient collective outcomes. As a result, whether and how voluntary cooperation can be sustained within organizations\(^\text{1}\) has long been recognized as a high-stake research question in organizational behavior, spanning the disciplines of economics (Groves et al., 1977), sociology (Marwell and Ames, 1979), political science (Olson, 1971) and social psychology (Dawes, 1980).

This literature originally relied on laboratory experiments in order to identify the individual preferences that may support voluntary cooperation. One of the most robust findings from this broad experimental literature is that cooperation gradually decays in repeated experiments (see Ledyard (1994) for an extensive review). This result can be explained by the fact that many individuals exhibit reciprocal preferences: they are “conditional cooperators” willing to cooperate as long as others do so as well. However, because reciprocators have a preference for matching the contributions they observe or expect from others, negative interactions can become self-reinforcing and increase the chances of organizational failure, consistently with the theory of reciprocal exchange (Gouldner, 1960; Cropanzano and Mitchell, 2005; Falk and Fischbacher, 2006). Cooperation is thus highly fragile and can only be sustained with specific institutional designs in the lab (Fischbacher and Gächter, 2010).

Paradoxically, this fragility of cooperation observed in the lab has yet to be confirmed in field

\(^1\)By “organization”, we mean any group or team of individuals who work together towards some shared purpose. This could be a nation, a firm, a not-for-profit organization, or an open innovation community (e.g., a research community).
organizations. Since reciprocal preferences are hard to identify in the field, recent empirical studies increasingly run “lab-in-the-field” experiments within real-world organizations (Gneezy and Imas, 2017). The spirit of lab-in-the-field experiments is to match individuals’ behavior in experimental games (the lab) with choices by the same individuals in the field and the performance of organizations they work in (Felin et al., 2015; Bitektine et al., 2018). Over the past decade, this growing literature has yielded stark empirical results: groups composed of more reciprocal types, as measured by incentivized lab experiments, achieve significantly better outcomes in a variety of settings, such as common forest management (Rustagi et al., 2010), fisheries (Carpenter and Sekli, 2011), profit maximizing firms (Barr and Serneels, 2009) or financial markets (Anthony, 2005). This unambiguous relationship between reciprocity and group-level success in the field is surprising in light of the lab evidence, where reciprocal preferences drive cooperation levels down whenever conditional cooperators observe (or believe) that some team members are not doing their “fair share.”

Our paper makes two main contributions to the literature. First we highlight a possible limitation of the lab-in-the-field literature that may mechanically produce the observed unambiguous positive relationship between reciprocity and organizational success. The key to lab-in-the-field studies is contacting the lab participants within their work environment. We argue that this introduces a bias, at both individual and organization level. Failed organizations will simply not be sampled, since the members are no longer present and can
no longer be contacted. More generally, for organizations that have not yet died, 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 where an inability to sustain cooperation can more easily lead to the death of the organization (i.e., private firms or voluntary communities, as opposed to, e.g., a government agency), in particular for young and fragile organizations.

To address this concern, we propose an alternative way of using lab data in the field. The lab data can be used to measure reciprocity motives. We then propose a field measure of reciprocity (inspired by the experimental game) and show that the lab and field measures strongly correlate. Finally we use our field measure combined with a large panel dataset, including dead organizations, to test whether organizations with a large proportion of reciprocators are indeed those that fare better.

We conducted this study in the context of Sourceforge, a large online platform hosting virtual teams that develop open source software (OSS) projects. This environment is particularly well-suited to our study. Individual projects largely rely on voluntary contributions from team members in order to thrive. At the same time, these projects are made freely available for anyone to use. Taken individually, team members therefore face a repeated public good dilemma where they get to observe teammates’ past contributions to the project, and have to decide how much they want to contribute in each period, if anything. Additionally, the sampling bias can be particularly severe in this innovation-driven environment, where open source development projects are quite likely to fail.

In 2011, we contacted 2534 open source developers registered with Sourceforge. The pool of developers we contacted was constructed using as stratifying variables the size of the projects as

---

2See section 3.1 for more details on how open source software works.
well as the license type (Belenzon and Schankerman, 2015). We eventually collected lab data on
1194 experimental subjects, and use their conditional contribution decisions in the public goods
game to measure reciprocity in the lab.

In parallel, we collected field data on the universe of projects our experimental subjects and
their collaborators had been contributing to, yielding a final sample of 5557 projects involving
10537 developers overall. For each project, we extracted the monthly number of code contri-
butions (or “commits”) made by every developer, as well as all available project-level informa-
tion: age, license type, programming languages used, natural languages used at the team level,
and target user population. We also separately extract from Sourceforge its flagship measure of
project-level success – featured prominently on each project summary page – called the “activity
percentile”. We use this measure as an exogenous proxy for success at the team level.

The panel data dimension allows us to achieve two main objectives. First we exploit the dy-
namics of contributions to build a field measure of reciprocity, inspired by the lab measure. We
compute the correlation between a developer’s own contributions to a given open source project
and the sum of contributions made by his or her fellow team members in the previous period.
The idea is to measure the individual’s reaction to teammates’ choices. To overcome the possibil-
ity that project-level shocks (e.g., productivity shocks) could drive the contributions of both the
developer and his or her co-workers, we exploit the fact that the team members of a given de-
developer typically contribute to several unrelated projects at the same time, allowing us to identify
exogenous variations in their contribution levels. We then look at how each developer reacts to
those exogenous variations in team members’ past project contributions to derive a measure of re-
ciprocal preferences at the individual level. We show that field and lab measures of contributions
are strongly correlated.

Second, using this field reciprocity measure, we make our second main contribution. We show
that, consistent with lab results, the relationship between reciprocal preferences and organizational success is indeed ambiguous (at least in our setting). Organizations that have a higher share of highly reciprocal members are significantly more successful on average (based on both lab and field measures of reciprocity), but that this is conditional on surviving. This result is consistent with the previous lab-in-the-field literature. When considering our full sample of teams, however, we find no relationship between reciprocity and success levels. In fact, teams that have a higher share of reciprocators are more likely to fail and disappear. Consistent with existing models of reciprocal exchange in organization science, social psychology and economics, we show empirically that the mechanism behind those aggregate results is that reciprocity functions as a catalyst. It reinforces contribution dynamics when the team is already mobilized, but also accelerates slowdowns when members perceive that others are not contributing – leading to a lower probability of recovery after a period of inactivity (the project dies). This problem might be particularly pervasive in complex and/or virtual work environments, where individual effort is harder to observe and low output due to idiosyncratic shocks can be interpreted as a lack of goodwill.

Our results are robust to various definitions of project death and field reciprocity (see Appendix C). Importantly, we can rule out the possibility that our field measure of reciprocity picks up other individual-specific characteristics (e.g., intellectual ability) by showing that the link between reciprocity and success (or death) vanishes when we focus on the sub-sample of single-authored projects. Finally, we can also exclude the interpretation according to which dead projects would in fact be the most successful ones – having reached a mature stage where 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 (these results are reported in Appendix B).

The rest of the paper proceeds as follows. Section 2 discusses the relevant related literature.
Section 3 provides some background on the OSS environment, and details our data collection strategy. Section 4 defines our laboratory and field measures of reciprocity preferences, and discusses their correlation. Section 5.1 reports our main results on the relationship between reciprocity and success at the team level, while section 5.2 leverages the panel structure of our data to provide evidence on the behavioral mechanism posited behind those aggregate results. Section 6 concludes.

2 Related literature

The early literature attempting to identify individual preferences, beliefs and institutional designs encouraging voluntary cooperation relied on laboratory experiments (Ledyard, 1994; Roth, 1995). One of its most robust findings is that cooperation gradually decays in repeated experiments. While most individuals initially make non-zero contributions, their willingness to cooperate declines with repetition (see Chaudhuri (2011) for a survey). This finding may follow 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 will inevitably collapse if either: 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 less than others on average); iii) some group members have reasons to believe that others will reduce their contributions in the future.

The major takeaway from this experimental lab-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 which either discipline non-reciprocating types (through monetary and non-monetary pun-
ishment), or provide a mechanism for excluding them from the group (Fehr and Gächter, 2000; Chaudhuri, 2011; Bartling et al., 2012).

More recently, the literature has turned to the lab-in-the-field approach to determine whether the predictions obtained in the lab are borne out in real-world environments. In particular, the literature has focused on how reciprocal preferences, and cooperation more generally, relate to real-world outcomes (see Gneezy and Imas (2017), Levitt and List (2009)). This fast-growing literature relies on validated lab paradigms to elicit psychological traits or preferences and connects these measures with outcomes of theoretical interest in the field.

The initial lab-in-the-field studies matched behavior in the lab 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 Vileval (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, respectively, and show that more cooperative subjects are less likely to engage in over-extraction, 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.3

3There 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 where 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 subjects – which corresponds to the way chiefs are appointed in this field setting. Gilligan et al.
However, there are fewer lab-in-the-field studies on the link between reciprocal preferences and group-level outcomes. The first paper to study this question in a real-world setting was Anthony (2005), who did not adopt a lab-in-the-field approach per se, but rather relied on survey answers. They measure reciprocal behavior in 106 microcredit borrowing groups in the U.S. through a number of survey questions answered by randomly selected group members. The paper finds reciprocity to be the variable most strongly associated with low levels of loan delinquency, and higher group longevity. Barr and Serneels (2009) were the first to obtain experimental measures of reciprocal preferences from a sample of workers in 20 Ghanaian manufacturing firms, which they couple with survey-based data on workers’ individual wages, and aggregate firm productivity. They find that reciprocal workers earn higher wages on average, as well as finding a strong firm-level relationship between reciprocating behavior and aggregate productivity. Rustagi et al. (2010) study 49 local groups participating in a publicly funded forest conservation program in Ethiopia, where they are responsible for maintaining and growing the forest (the “public good”). They elicit reciprocal preferences with a conditional public goods game (Fischbacher et al., 2001), and notably measure the share of “conditional” and “weak conditional” cooperators in each group, which they relate to an independently collected measure of success in forest commons management. They find that groups with a larger share of highly reciprocating types are more successful on average. Similarly, Carpenter and Seki (2011) rely on the public goods game to elicit reciprocal preferences from Japanese fishermen. Even though their sample only contains 12 fishing crews, they find that those exhibiting higher levels of reciprocity are more productive on average. (2014) exploit 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) run lab-in-the-field experiments in Liberia to show that exposure to war-time violence increases governments’ ability to instruct citizens to make voluntary contributions to public goods.
This lab-in-the-field approach has been a useful tool for organizational researchers interested in micro-founding the processes that drive aggregate outcomes in the field (for an overview of the micro-foundations movement in organization science, see, e.g., Felin and Foss (2005); Felin et al. (2015); Bitektine et al. (2018)). The papers reviewed above send a similar message: groups composed of more reciprocal types, as measured by experimental games, are shown to achieve better aggregate outcomes in a variety of settings where individuals face social dilemmas. This unambiguous relationship between reciprocity and field group success is surprising in light of the lab evidence, where reciprocal preferences drive cooperation levels down whenever conditional cooperators observe (or believe) that some team members are not doing their “fair share.”

We contribute to this broad lab-in-the-field literature in two ways. First, we show that high reciprocity can also lead to organizational failure. We argue that such negative dynamics are not captured by existing lab-in-the-field studies, because the cross-sectional nature of their data implies that only surviving teams can be observed. Second, the panel nature of our data allows us to demonstrate the micro-level psychological mechanism behind our aggregate results: reciprocity reinforces positive work dynamics but also negative ones, so that a lack of contribution in one period can be self-reinforcing and lead to the decay of cooperation at the team level. One direct implication of our results in this respect is that field cooperation can collapse and lead to the death of an organization even when it comprises nothing but highly reciprocating types. A sufficient condition for the unravelling of cooperation within a group of strong reciprocators is that some receive a signal that others are not contributing in one period – even if this signal does not accurately reflect actual effort levels.

At a theoretical level, our paper is related to the literature on social exchange theory (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. While
our paper is distinct from the analysis of the social networks structure of organization discussed in the sociology variant of the SET literature (Cook and Whitmeyer (1992), Willer et al. (2012)), 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 should feel bound to respond in kind. 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 has relied on laboratory experiments to test these predictions (see, e.g., 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 a real-world, online generalized exchange system such as OSS, where peers intend to co-produce a public good based on voluntary contributions. We exploit the fact that OSS teams differ in the extent to which their members endorse reciprocity, in such a way that we can 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, where researchers can gather a significant amount of field data on real-world behavior (Lazer et al. 2009), while retaining the ability to conduct controlled

---

4The formal definition of reciprocity is strikingly similarly in economics (see, e.g., Sobel 2005, Dufwenberg and Kirchsteiger 2004 and Falk and Fischbacher 2006). Further, 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), had previously been established in the context of lab-based tests of SET (see, e.g., Eisenberger et al. 1987).
experiments (Hergueux and Jacquemet, 2015).

Finally, we stress 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) note that SET actually “fails to distinguish the presence of negative constructs (e.g., abuse) from the absence of positive construct (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), while 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 downwards, and eventually stop contributing. Since the ability to “punish” non-reciprocating 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 lab designs where 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 Page et al. (2005), Cinyabuguma et al. (2005) and Charness and Yang (2014)).

\[5\] In their review, Cropanzano et al. (2017) argue that positive and negative reciprocity preferences might in fact not be strongly correlated within subjects. To the best of our knowledge, no experimental design in the literature on public goods provides subjects with both the ability to exclude and/or punish other group members based on past contribution behavior. As a result, “punish” could be selected as a second best response by some conditional cooperators in designs \(\text{à la} \) Fehr and Gachter (2000), where they would in fact prefer to break the relationship (e.g., “leave”).

\[6\] Note that the possibility of decentralized punishment does not necessarily lead to enhanced cooperation and welfare. In designs where punished subjects have the possibility to “counter-punish,” many decide to engage in welfare reducing “anti-social punishment” (Denant-Boemont et al. 2007; Nikiforakis 2008; Nikiforakis et al. 2012; Herrmann et al. 2008).
3 Setting and data collection

3.1 Open source software

To set the stage, we provide some background information on open source software. OSS currently engages 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 abovementioned projects, which are both very large and quite famous, 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 freely provide developers with a set of 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, pre-specified design rules, or markets [Benkler, 2002; Hippel and Krogh, 2003; Von Krogh and Von Hippel, 2006]. Contributors typically resolve potential disagreements over future developments through discussion, and, in some rare case, 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”
where 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 which are made freely available for anyone to use, OSS has been described as a privately-produced public good (O’Mahony, 2003), where developers reveal their code in the expectation that others will reciprocate (Maurer and Scotchmer, 2006). About 50% of OSS developers are pure volunteers who contribute in their free time, with the other half deriving either direct or indirect revenue from their contributions (Hertel et al. 2003; Lakhani et al. 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 dedicate working hours to any project of their choosing, on the assumption that developing OSS will (i) allow them to acquire new skills, and (ii) keep them in touch with a fast-moving open innovation community.

3.2 Collecting lab data

In May 2011, 2534 OSS developers registered with Sourceforge.net were contacted 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 1194 subjects – a 47% take-up rate – participated. Before describing the details of the experimental procedure, we start by describing how the initial sample of 2534 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 below. There is great heterogeneity between Sourceforge projects in terms of number of contributors, and previous research efforts have been somewhat biased towards a handful of large and highly successful projects (Crowston et al., 2012). To avoid this pitfall, the first stratification variable that 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.

Specifically, we extracted from Sourceforge all the projects that were active in 2010, defined as having either a bug closed or a last feature added in 2010. This yielded a sample of 1577 active projects. After excluding the projects for which the SVN logs – i.e. the logs to the software revision control system which provides detailed information on code contributions at the developer level, information necessary to conduct the analysis – were inaccessible, we were left with a sample of 1242 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 (2015)’s classification of the 44 existing OSS license types to label their licensing terms as highly, moderately or weakly restrictive.

---

*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).*
Since there were only 83 projects with more than 7 active contributors, we selected all of these projects irrespective of their license terms. For all the projects with 6 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 projects featured highly restrictive licenses, 57 projects 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 size.

For the 322 projects we selected, we kept all 1019 developers who were active in 2010. In addition, we also randomly selected 3 non active developers. We ended up with a sample of 2534 Sourceforge developers eligible to participate in the experiment. Table 1 summarizes the selection procedure.
With the support of the Sourceforge platform, we collected the e-mail addresses of all 2534 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 uniquely identify them and subsequently collect their entire history of contributions to OSS. Subjects were then redirected to the welcome screen of the experimental platform.

Given that this was an online experiment, we needed a fully self-contained interface. The welcome page of the decision interface provided subjects with general information about the procedures. Our design strictly follows the experimental procedures detailed in Hergueux and Jacquemet (2015). These procedures have been developed specifically to strengthen the internal validity of Internet-based experiments. Their reliability was established through a careful comparison of decisions elicited in the lab and online.
experiment, including the number of sections, expected completion time and how earnings are computed. In order to minimize potential demand effects and in-group biases, we were very careful not to present the study as OSS-oriented. We made it very clear on the introductory screen that subjects would interact with a diverse pool of Internet users. Final earnings were computed by randomly matching our subjects with individuals from a pool made up of OSS developers, Wikipedia users and students.

After this introduction, the participants played a conditional public goods game, which we describe in more detail below. At the end of the experiment, we asked subjects for some standard demographic information, i.e. their age, gender, education and salary range. We also asked them a few survey questions on their motivations for contributing to OSS. We used these variables as controls in our regression analysis.

We find that the population of OSS developers who answered our survey is young on average (32 years old) and overwhelmingly male (only 3% of developers are female). The average developer in our experiment has a 4-year college degree (BA, BS), with 17.5% of the population of developers having a lower qualification than a 2-year college degree and almost half of the population (i.e. 49%) having a Masters degree or a PhD. The average developer earns between $2000 and $4000 per month, with 32% of the population earning less than $2000 and 20% earning more than $7500. These statics are consistent with survey studies on OSS developers (see, e.g., David and Shapiro (2008)).

9In addition, all subjects were subsequently presented with other decision tasks which we do not exploit in the context of this paper. These were presented in varying order after the public goods game, and included a dictator game, an ultimatum bargaining game, and a trust game. 10Specifically, we followed the previous literature on OSS (see, e.g., David and Shapiro (2008)) and asked subjects to state their level of agreement with the following reasons for contributing to OSS: (i) because I think software should not be a proprietary product, (ii) because I like to learn and develop new skills, (iii) because I need to solve a problem that could not be solved by proprietary software, (iv) because I want to get a reputation on the OSS developers scene, and (v) to make money.
and Shapiro (2008)).

The key experimental game we used to elicit reciprocal preferences is the one-shot public goods game. This game is played in groups of 4 players, each with an initial endowment of 10 dollars. Group members need to decide how much to contribute to a common project. Each dollar invested in the common project produces 1.6 dollars, which is then equally distributed among group members. Thus, a one-dollar investment only yields a private return of 0.4 dollar, 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 $\text{contrib}_i$, the final private payoff is given by:

$$\pi_i = 10 - \text{contrib}_i + 0.4 \sum_{j=1}^{4} \text{contrib}_j.$$ 

Following Fischbacher et al. (2001), we elicited two types of contribution decisions: first an unconditional contribution, then a conditional contribution. For the unconditional contribution, each subject had to decide on his or her contribution in the game described above. For the conditional contribution, each subject determined his or her intended contribution for each possible value ($0, 1, 2, \ldots, 10$) of the average contribution of the other three members of the group. The conditional contributions allowed us to measure subjects’ willingness to behave reciprocally (i.e., be conditionally cooperative). This design is incentive-compatible since, after the match with other participants has been realized, one randomly selected decision (i.e., unconditional or conditional) is used to compute subjects’ earnings. The screen eliciting conditional contributions is presented

11 We used a one-shot design because a repeated game would introduce strategic concerns that would obscure the interpretation of subjects’ behavior.

12 More specifically, two group members are randomly selected to make an unconditional contribution. The contribution of the remaining two group members is then determined based on their conditional decision, according to the average of the unconditional contributions.
One important methodological aspect of the online implementation of the experiment is to guarantee a quick and appropriate understanding of the instructions when no interaction with the experimenter is possible. We strengthened the internal validity of our online experiment through three distinctive features of the interface. First, we included novel flash animations illustrating the written experimental instructions at the bottom of the instruction screen (see Figure 2). Second, the loop of concrete examples displayed in each animation was first randomly determined and then fixed for each game. The same loop was displayed to all subjects without any other numeric information than the subject’s initial endowments. We decided against displaying a purely random sequence of flash animations as it could have introduced uncontrolled and subject-specific noise-through, e.g., anchoring on a particular behavior or sequence of events. Our goal with these animations was to illustrate the basic gist of each decision problem in an accessible way while preventing specific numerical examples and results from predominating in subjects’ minds.
the instruction screen was followed by a screen providing some examples of decisions, along with a detailed calculation of the resulting payoffs for each player. These examples were supplemented on the subsequent screen by an earnings calculator. On this interactive page, subjects were allowed to test any scenario they wanted to consider. Last, the system provided quick access to the instructions material at any moment during decision-making.

**Figure 2: The instruction screen of the Public Goods game**

![Image of the instruction screen of the Public Goods game](image)

Final payoffs were computed using the earnings from one randomly selected decision task. In addition, the players obtained a $10 participation fee. Final payments were made via an au-
tomated PayPal transfer\textsuperscript{14}. It is important to stress that OSS developers can be very hostile to monetary rewards. In order to ensure that the experiment was equally incentive-compatible for all subjects, we allowed them to donate their final earnings to the International Committee of the Red Cross upon completion of the experiment. This possibility was made clear on the welcome screen of the decision interface.

3.3 Field data

In addition to the lab 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 (SRDA)\textsuperscript{15}, 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, when Sourceforge put an end to its data sharing agreement.

We collected this data for all team members belonging to (i) projects to which our starting set of 1194 experimental subjects contributed, and (ii) all the other projects their teammates (9343 co-developers) worked on without the subjects’ participation. We obtained a final sample of 5557 OSS projects involving 10537 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

\textsuperscript{14}Such a payment procedure guarantees a fungibility similar to that of cash transfers in lab experiments, as money transferred via PayPal can be readily used for online purchases or easily transferred to one’s personal bank account at no cost. We only required a valid e-mail address to process the payment. To strengthen the credibility of the payment procedure, we asked subjects to enter the e-mail address that was (or would be) associated with their PayPal account right after the introductory screen of the decision interface.

\textsuperscript{15}See http://Archive, www3.nd.edu/oss/Data/data.html
In addition, we extracted the creation date of each project, and exploited the fact that OSS development teams often document the characteristics of their projects on the Sourceforge platform to collect additional project-level information, including license restrictiveness\(^\text{17}\) popularity of the programming languages used\(^\text{18}\) natural languages used\(^\text{19}\) and target user population\(^\text{20}\). We used these variables as controls in our project-level regressions.

Last, we needed to address the challenge of reliably measuring the level of success of OSS projects. Indeed, since 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, since 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 (Grewal et al., 2006; Crowston and Scozzi, 2002; Crowston et al., 2004; 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 above 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) put it, “projects with high activity percentile are popular projects since it is based on downloads, site views, development activity, and administrator activity.”

\(^\text{16}\) 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).

\(^\text{17}\) This variable ranges from 1 (low restrictiveness) to 3 (high restrictiveness), as in Belenzon and Schankerman (2015).

\(^\text{18}\) We took the log number of teams in our dataset that report the use of any given programming language as a measure of overall programming language popularity. We then computed the average of these popularity indicators at the project level (since many teams use several programming languages at the same time).

\(^\text{19}\) This variable equals 1 if the team lists English as a working language, 0 otherwise.

\(^\text{20}\) This variable equals 1 if end users are listed as the target population for the software, 0 otherwise.
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:

\[
\text{Activity Percentile} = \frac{1}{3} \text{User Traffic} + \frac{1}{3} \text{Development Activity} + \frac{1}{3} \text{Project Communication}.
\]

4 Measuring reciprocity

As explained in the introduction, lab-in-the-field experiments typically adopt the following methodology. At the time of the experiment, a sample of existing organizations is selected, the members of these organizations participate in experimental games (the lab 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 (i.e., less likely to die according to performance), this is a powerful tool.

However, in the context of OSS as in many others (e.g., private firms or volunteering organizations), organizations are much more volatile: some might grow, while others may quickly fail and die. Thus, when the experiment is run, only the participants in projects that have not died are available to be sampled. More generally, the activity level of a project, and thus the effective presence of its members on the platform, may impact the likelihood that they participate in the experiment. This would then generate sampling bias, where successful organizations are over-represented compared to those that fail.

While it could have been interesting to decompose this indicator for the purpose of our analysis, Sourceforge does not provide the disaggregated components of its activity percentile metric, and the variables that go into its computation cannot be retrieved. The measures are defined as follows:

\[
\text{User Traffic} = \frac{\ln(1+\text{total downloads})}{\ln(1+\max[\text{all projects}])} + \frac{\ln(1+\text{total logo hits})}{\ln(1+\max[\text{all projects}])} + \frac{\ln(1+\text{total website hits})}{\ln(1+\max[\text{all projects}])},
\]
\[
\text{Development Activity} = \frac{\ln(1+\text{total commits})}{\ln(1+\max[\text{all projects}])} + \frac{100-\text{nb days since last file release}}{100} + \frac{100-\text{nb days since last project admin login}}{100},
\]
\[
\text{Project Communication} = \frac{\ln(1+\text{total bug tracker submissions})}{\ln(1+\max[\text{all projects}])} + \frac{\ln(1+\text{total mailing list posts})}{\ln(1+\max[\text{all projects}])} + \frac{\ln(1+\text{total project forum posts})}{\ln(1+\max[\text{all projects}])}.
\]
To overcome this problem, we used our lab data in a different way. The lab data served to validate an analogous field measure of reciprocity, which we could then exploit in conjunction with our panel data so as to expand our sample to include team-members and projects that were already dead at the time of our study. Below we describe our lab measure, then discuss the equivalent field measure that we propose, before showing how the two measures correlate.

4.1 Experimental measure of reciprocity

As described in Section 3.2.2, our subjects in the public goods game reported both unconditional and conditional contribution decisions. As in Fischbacher et al. (2001)’s seminal paper, we used the conditional contribution decisions to compute a measure of reciprocal preferences at the developer level. We defined this measure as the correlation between the player’s conditional contribution decisions and the corresponding average contribution of the three other members of his or her group (from 0 to 10 dollars, as illustrated in Figure 1). This variable was distributed with a mean of 0.73 and a standard deviation of 0.45. Note that it can only capture positive reciprocity, not negative reciprocity, since in the standard public good games we used, there was no opportunity to pay a personal cost to decrease others’ earnings.  

While directly inspired by Fischbacher et al. (2001), the experimental measure of reciprocity that we defined was computed as a simple correlation, which captures subjects’ willingness to be conditionally cooperative in the public goods game (i.e., behave reciprocally). This measure has the benefit of simplicity. It also had a direct analog in our field setting, as we explain in the next section. By comparison, Fischbacher et al. (2001) classify their student subjects in 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 would imply

22The worst you can do to hurt others is to contribute 0, which is also the best for your own private payoff.
a correlation of zero in our setting), (ii) conditional cooperators, who match the contributions of the other members of their group (this would imply a correlation of 1 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 less than others on average (this would imply a positive correlation of less than 1 in our setting).  

4.2 Generalizable field measure of reciprocity

The next step in our study was to propose a field measure of reciprocity, inspired by the lab 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 his or her team members in the previous month. This measure might however be biased, since both the contributions of a given developer and those of his 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 affect the contribution level of a developer and those of his or her team members in the previous period. We might thus incorrectly conclude, based on a positive correlation, that the individual was reciprocal.

We thus proposed and built a field measure of reciprocity that corrected for this concern. For each developer, we computed the correlation between his or her contributions at the project ×

---

23Some differences between our pool of OSS developers and the populations of students typically used in lab experiments are noteworthy. Only 4% of our subjects could be classified as free-riders (as compared to 20-30% in student populations), 48% were perfect reciprocators, and 41% 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).
month level and the predicted contributions of his or her fellow team members in the previous month. This predicted measure used the variation in the sum of contributions made by their own collaborators on the other projects that they pursued independently. By “independently”, we mean that we required that the developer under consideration did not himself contribute to those other projects, and thus never directly interacted with the collaborators of his or her team members. We provide the formal description of the measure in Appendix A.

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 his/her team members $j$ in $t - 1$, as predicted by the exogenous variation in contributions of their own team members $k$ in $t - 2$ on the projects which they pursue without developer $i$’s involvement.

When computing the measure this way, we find that 3700 out of the 8250 developers for whom we can compute a field measure of reciprocal preferences in our final sample have a measure of reciprocity which 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).
These could for instance be “altruistic” developers who care about providing as much public good as possible, so that 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 design, where subjects are faced with a single common project. Alternatively, this pattern could fit with a story 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 lab measures, where participants cannot contribute negatively to the public good, we defined our field measure of reciprocity as the maximum of zero and of the correlation calculated above.

Table 2 shows that, regardless of the individual level controls we introduced, we find that our field and lab measures are strongly correlated. When the lab measure increases from 0 to 1, the field measure increases by 0.5 on average.

<table>
<thead>
<tr>
<th>TABLE 2: CORRELATION BETWEEN LAB AND FIELD MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) (2) (3)</td>
</tr>
<tr>
<td>reciprocity field</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>reciprocity lab</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Socioeconomic controls</td>
</tr>
<tr>
<td>Reported motives controls</td>
</tr>
<tr>
<td>R-squared</td>
</tr>
<tr>
<td>N. of obs</td>
</tr>
</tbody>
</table>

Column (1) includes no control, column (2) adds gender, age and income, column (3) adds reported motives for contributing to OSS.
5 Reciprocity and the success or failure of organizations

In the last part of the paper, we show how correcting for sampling bias affects the results on the role of reciprocity in the success of organizations. We present these results in Section 5.1 and analyze the underlying mechanism in Section 5.2.

5.1 Reciprocity and failure

We set the stage in Table 3, which relates the mean of our experimental measure of reciprocity at the team level to average project-level success. In all our analyses, we standardize the activity percentile variable, so that the estimated coefficients represent the effect of reciprocity in terms of standard deviations of the success score in our population of projects. We see that, 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 by the previous literature. Regardless of the controls we include, there is a statistically significant link between reciprocal preferences at the team level and organizational success.

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>success score</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(mean) reciprocity</td>
<td>0.12∗∗</td>
<td>0.13∗∗</td>
<td>0.16∗∗∗</td>
</tr>
<tr>
<td>(0.06)</td>
<td>(0.06)</td>
<td>(0.06)</td>
<td></td>
</tr>
<tr>
<td>R-squared</td>
<td>0.00</td>
<td>0.26</td>
<td>0.25</td>
</tr>
<tr>
<td>N. of obs</td>
<td>1143.00</td>
<td>1013.00</td>
<td>933.00</td>
</tr>
</tbody>
</table>

Robust standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

Column (1) includes no controls, column (2) adds gender, age and income, column (3) adds reported motives for contributing to OSS.

The rest of the paper then shows that, if we use our field measure of reciprocity on our ex-
panded sample of organizations, we get much more nuanced results. We start 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. For these three different types of projects we plot the proportion of high reciprocators, defined as those that have a field reciprocity measure above the 75th percentile. Consistent with previous lab-in-the-field evidence, Figure 4 shows that low-success projects have 40% of high reciprocators in their team on average, while high-success projects have 50% of high reciprocators – a 25% increase in proportion. Strikingly, however, dead projects do not differ significantly from highly successful ones in terms of the share of high reciprocators in their teams: 52% on average.

24For 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.
This graphical analysis is confirmed in a regression framework. In Table 4, we examine the relationship between the proportion of high reciprocators in a given OSS team and project-level performance. All regressions control for project-level characteristics (age, license restrictiveness, popularity of the programming languages used, natural languages used at the team level, and target user population), and rely on robust standard errors for inference. In column (1) the dependent variable 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 this social type is associated with a 12% increase in the probability of project failure and death.

Next, we examine how the proportion of high reciprocators influences team success. Column (2), using the activity percentile as the dependent variable, shows that groups with a high share of
high reciprocators 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 in the full set of projects. This relationship becomes stronger in column (3), where we restrict the analysis to organizations that did not disappear over the period.

In columns (4) to (6) we reproduce columns (1) to (3), but add the restriction that we can compute our field measure of reciprocity for at least one third of team members, so as to increase confidence in our estimate of the share of high reciprocators. With this restriction, we see that for the full set of projects, the relationship between the share of high reciprocators and success is no longer statistically significant (column (5)). We recover a strong positive relationship when we restrict our sample of projects to those that did not fail over the period (column (6)).

<table>
<thead>
<tr>
<th>TABLE 4: SUCCESS AND SURVIVAL: EXPANDED SAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>share of high reciprocators</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>R-squared</td>
</tr>
<tr>
<td>N. of obs</td>
</tr>
</tbody>
</table>

Robust standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

Column (1) is an OLS regression of the dummy variable taking the value 1 if the project died during our time period. Columns (2) and (3) are an OLS regression of the success score of a project on the share of high reciprocators in the project. Column (3) is restricted to projects that survive during the entire time period. All regressions include project-level controls for project age, license type, the popularity of the programming languages used, the natural languages used and target user population. We restrict our sample to projects with at least 2 developers. Columns (4) to (6) reproduce columns (1) to (3) but are restricted to projects where we have our field measure of reciprocity for at least one third of members.

Our results therefore show that sampling bias is likely to have prevented previous lab-in-the-field studies from replicating the lab-based findings that reciprocal preferences can also lead to
failure at the team level. In Appendix B we consider a number of robustness checks on these results. One concern could be particularly worrisome, and so we address it in the main text. The issue could be that our field measure of reciprocity might capture individual-specific characteristics other than reciprocal preferences, such as the developer’s cognitive abilities. We show that this is unlikely.

In Table 5, we reproduce the first three columns of Table 4 but restrict the sample to projects with a single developer, where reciprocity preferences cannot drive the dynamics of cooperation. When we do so, we fail to find any strong statistical link between each developer’s level of reciprocity and either the success of his project, or the probability that it eventually fails and dies. This suggests that our results are not driven by omitted variable bias.

<table>
<thead>
<tr>
<th>TABLE 5: PROJECTS WITH A SINGLE PARTICIPANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>share of high reciprocators</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>R-squared</td>
</tr>
<tr>
<td>N. of obs</td>
</tr>
</tbody>
</table>

This table reproduces the first three columns of Table 4 but restricted to projects with a single participant.

5.2 Mechanism: the reciprocity rollercoaster

In the lab, reciprocal preferences typically work as a catalyst: they amplify positive and negative contribution dynamics within teams equally. As a result, the presence of many strong reciprocators in a team will drive the decay of cooperation when some group members make relatively small contributions in one period (Fischbacher and Gächter 2010). At the same time, it sustains very high levels of cooperation when everybody contributes at a relatively high rate (Page et al.)
Our paper is the first to provide an empirical test of this micro-level mechanism within real-world organizations. We hypothesize that high reciprocators will react positively to increased contributions from team members, but will 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 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 members to believe that others are not contributing. This is especially important in real-world contexts, where realized contributions (i.e., contributions that are observable to other team members) can imperfectly reflect individual effort levels. Over the course of many months or years, some periods of inactivity in our virtual team setting will likely result from idiosyncratic shocks at the individual or project level. Such shocks could then drive beliefs about individual effort levels, and ultimately determine the dynamics of cooperation within the team.

In table 6, we rely on the dynamic nature of our data on individual team contributions to analyze the dynamics of contributions within projects at a monthly frequency over the eight years period that we cover. To do so, we run individual panel regressions where our unit of observation is at the developer x project x month level.

We first analyze whether the share of highly reciprocal developers at the team level impacts the probability of the project recovering from a period of inactivity (columns (1) and (2)). In both columns, 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 in column (1) is a dummy indicating whether the developer made
any contribution to the project in the following month. This regression controls for all available project-level characteristics (similar to table 4), with robust standard errors clustered at the project level. Facing a period of inactivity increases the probability of the developer making no contribution in the subsequent month by 13% when he or she 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 an 11% lower probability of making no contributing in the following period.

These results are confirmed in column (2), where we take the total number of contributions made by a developer to a project in a given month as an alternative dependent variable. The specification is similar to that of column (1), except that we now include developer fixed effects in order to properly account for unobservable characteristics at the individual level. We obtain similar results. Facing a period of inactivity decreases the average number of contributions made in the current period by 1.21 for non reciprocators. In the case of high reciprocity types, this number decreases to -3.6 – a threefold increase in magnitude.

Column (3) relies on the same econometric specification as column (2), but focuses on the benefits of positive reciprocity at the team level, i.e., the fact that reciprocal exchange dynamics should reinforce positive development trends. 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 on average 2.13 more contributions following a period of high activity, while highly reciprocal ones make on average 5.33 more contributions – a 150% increase.

\[25\] The inclusion of developer fixed effects explains that the “high reciprocity type” coefficient is dropped from the estimation in columns (2) and (3).
in contribution levels. Taken together, these results provide evidence in support of the micro-level mechanism driving our aggregate result that, all else being equal, reciprocity can increase either the probability of death or the level of success of organizations.

<table>
<thead>
<tr>
<th>TABLE 6: MICRO-LEVEL MECHANISM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>no contribution last 3 periods</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>interaction with high reciprocity type</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>high reciprocity type</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>above median contributions last 3 periods</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>interaction with high reciprocity type</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>R-squared</td>
</tr>
<tr>
<td>N. of obs</td>
</tr>
</tbody>
</table>

Robust standard errors clustered at the project level in parentheses

* p<0.1, ** p<0.05, *** p<0.01

In column (1) we regress the dummy variable taking a value of 1 if no contribution was made by the participant in a given month and for a given project on a variable measuring whether no contributions were made in the previous 3 months and an interaction of this variable and the fact the individual is a high reciprocator. In columns (2) and (3) we use as a dependent variable the number of commits. All regressions include project-level controls and columns (2) and (3) include individual fixed effects.

6 Conclusion

The existing lab-in-the-field literature emphasizes that reciprocity is an important mechanism through which cooperation can be sustained and higher success achieved within organizations.
The message that this literature delivers to organizational scholars is straightforward: organizations should seek to engage reciprocal individuals. This unambiguous result is surprising in light of the extensive laboratory literature on voluntary cooperation, where positive reciprocity preferences drive the decay of cooperation in repeated public goods experiments (Fischbacher et al., 2001; Fischbacher and Gächter, 2010; Chaudhuri, 2011), and leads to above average performance only when highly reciprocal types manage to maintain relatively high cooperation levels (through, e.g., sorting (Page et al., 2005; Cinyabuguma et al., 2005; Charness and Yang, 2014)).

The lab-in-the-field methodology has many advantages for organizational research. The experimenter can rely on validated lab paradigms to elicit psychological traits or preferences “in the wild”, and link them to outcomes of theoretical interest in the field. In this process, he or she can maintain a tight 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 a powerful tool to disentangle competing micro-level mechanisms that could potentially drive aggregate empirical regularities, therefore contributing to the microfoundation movement in organization science (Felin and Foss, 2005; Felin et al., 2015; Bitektine et al., 2018).

The lab-in-the-field method also has some limitations. In this paper, we hypothesize that the relative disconnect between “pure” laboratory-based results and lab-in-the-field evidence stems from the fact that the latter methodology forces researchers to collect data from subsamples of surviving (and therefore relatively successful) organizations, which might bias their estimates of the aggregate relationship between reciprocal preferences and success. The more likely organizations are to disappear according to performance, the more severe the problem.

We thus conduct a standard lab-in-the-field experiment in the context of open source software development, where team members need to cooperate voluntarily towards the provision of a pub-
lic good. To correct for sample selection bias, we propose an alternative way to use lab data in the field, namely, to validate analogous field measures of psychological traits or preferences, allowing the researcher to use field “activity traces” to expand his or her research sample, notably in the direction of previously failed organizations.

Consistent with extant lab-based results, we establish that OSS teams that have a larger share of strong reciprocators are not necessarily more successful. In fact, they are significantly more likely to fail. We then exploit the detailed panel structure of our data to pinpoint the micro-level mechanism behind these aggregate results. Reciprocal preferences work as a catalyst at the team level: they increase contribution rates and effort levels whenever team members are strongly mobilized, leading to top-notch performance, but they also increase the probability that they leave the project altogether in periods of slow-down. Just like in the lab, positive reciprocity is a double-edged sword.

These results are relevant for those interested in using the lab-in-the-field methodology to study organizations, but are also important for the literature on the design and management of organizations themselves. Indeed, for reciprocal workers to stop cooperating, it is sufficient for them to believe that others are not doing their “fair share”, based on what they get to observe. In real-world organizations, observable individual contributions and input will usually imperfectly reflect actual effort levels. If team managers have better information than individual members about contributions and effort, they can decide to reveal or retain this private information so as to manage beliefs in a way that serves the purpose of cooperation.
References


Appendix

A Construction of the field measure of reciprocity

This appendix formally describes the construction of our measure of reciprocity in the field. Similar to our experimental design, the idea is to measure the contributions of an individual in reaction to variations in the aggregate contributions of his or her team members in the previous period. The main issue is that both contributions could be driven by a technological shock affecting the project, which would lead us to overestimate reciprocal preferences. In the spirit of an instrumental variables approach, we therefore measure the variation in the contribution level of each developer $i$ within our sample of 10537 developers at the project × month level as a function of the lagged contributions of his or her team members, which we predict using the (exogenous) variation in the contributions of their own team members on the projects which they pursue independently (i.e., where developer $i$ does not participate).

Specifically, consider developer $i$ working on a set of projects $\mathcal{P}_i$ at time $t$, for whom we want to measure reciprocity based on field data. For a given project $p \in \mathcal{P}_i$, we denote $y_{ipt}$ the contributions of that individual. An OLS specification of the relationship between the contributions of individual $i$ at time $t$ and the contributions of the other members of group $p$ at time $t - 1$ could be written as:

$$y_{ipt} = \beta_0 + \beta_1 \sum_{j \neq i} y_{jpt-1} + \beta_1 + \beta_p + \gamma_{ipt}$$

It would be natural to define $\beta_1$ as a measure of reciprocal behavior in the field.

However, in practice, because some projects might experience common productivity shocks, it
is likely that:

$$\text{corr} \left( \sum_{j \neq i} y_{jpt-1}, \gamma_{ipt} \right) \neq 0$$

We therefore construct a source of exogenous variation for $y_{jpt-1}$. To do so, we predict $y_{jpt-1}$ based on the variation in the contributions of developer $j$’s team members to the other projects to which he or she contributes at time $t - 2$, excluding those where $i$ participates (i.e. $p' \in P_j / p, p' \notin P_i$):

$$y_{jpt-1} = \alpha_0 + \alpha_1 \sum_{p' \in P_j / p, p' \notin P_i} \sum_{k \in p'} y_{kp't-2} + \alpha_j + \alpha_p + \epsilon_{jpt-1}$$

(1)

Columns (1) and (2) of Table 1 present the results of both the OLS and IV regressions. The contributions of $j$’s team members in his or her other projects at time $t - 2$ is a strong instrument for $y_{jp(t-1)}$ (the F-statistic reaches a value of 92). On average, if team members contribute more to projects $p' \in P_j / p, p' \notin P_i$, developer $j$ will, in turn, contribute significantly more to project $p$.

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ln(commits per month)$_p$</td>
<td>ln(commits per month)$_p$</td>
</tr>
<tr>
<td>Lagged contributions of team members in projects $p'$</td>
<td>0.09***</td>
<td>0.08***</td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td>(0.00)</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>N. of obs</td>
<td>9.9e+05</td>
<td>9.9e+05</td>
</tr>
</tbody>
</table>

Standard errors in parentheses

* $p<0.1$, ** $p<0.05$, *** $p<0.01$

For each developer, our measure of field reciprocity is therefore computed as the correlation between his or her contributions $y_{ipt}$ and the predicted value $\hat{y}_{jpt-1}$ of the contributions of his or her team members in the previous month, using $y_{kp't-2}$ as the explanatory variable, as specified in
equation If this correlation is negative, our reciprocity measure is constrained to take the value 0.

B Robustness

B.1 Measure of death of a project

A potential concern with our measure of death at the project level could be that it actually captures highly successful projects that have reached a mature stage where contributions are no longer needed. Two main arguments challenge this view. First, projects tend to die relatively more frequently at an early development stage, as can be seen from Figure A1. Second, as can be seen from Figure A2 the average number of commits that goes into each project tends to increase with its development stage, so that projects are actually most actively developed when they reach more advanced development stages.

Figure A1: Distribution of stage comparing dead and active projects
Finally, our main results are robust to restricting our sample to projects in early development stages. In Table A2, column (1) excludes projects that finish in production and mature stages, while column (2) further excludes projects that finish in beta version.

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>project dead share of high reciprocators</td>
<td>0.10*</td>
<td>0.16***</td>
</tr>
<tr>
<td></td>
<td>(0.06)</td>
<td>(0.05)</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>N. of obs</td>
<td>530.00</td>
<td>894.00</td>
</tr>
</tbody>
</table>

Moreover, we adopted a particular definition of dead projects: projects that received no contributions in the last 12 months of our sample. There is no official administrative data recording the death of a project and we chose 12 months as it both appeared reasonable and allowed us to
split our sample equally between dead projects (52%) and active projects (48%). We nevertheless examine the robustness of our results to changes in the definition, varying the number of months without contributions at the end of the project. In column (1) we consider a period of 6 months and in column (2) a period of 18 months. In all cases, an increase in the proportion of reciprocators in the group significantly increases the probability of death of the project.

<table>
<thead>
<tr>
<th>Table A3: Definition of Dead Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td><strong>Project dead (6)</strong></td>
</tr>
<tr>
<td>Share of high reciprocators</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>R-squared</td>
</tr>
<tr>
<td>N. of obs</td>
</tr>
</tbody>
</table>

Standard errors in parentheses
* p<0.1, ** p<0.05, *** p<0.01

One concern with our definition of dead projects is that projects could have migrated to a different platform. This is particularly a concern with the competing platform Github, which increased in prominence during our sample period. To address this concern, we collected information on the Github projects and, in the main specification of Table 4, removed all projects that migrated to Github. In table A4 we do not exclude these projects and show that the results are very similar, indicating that this was not in fact a major concern for our estimation.

---

26 The competitor launched by Google, called Google Code, never managed to challenge Sourceforge.
TABLE A4: RESTRICTING THE SAMPLE TO PROJECTS THAT DID NOT MIGRATE TO GITHUB

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>success score</td>
<td>project dead</td>
<td>success score</td>
</tr>
<tr>
<td>share of high reciprocators</td>
<td>0.05</td>
<td>0.17***</td>
<td>0.16*</td>
</tr>
<tr>
<td></td>
<td>(0.08)</td>
<td>(0.05)</td>
<td>(0.10)</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.22</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>N. of obs</td>
<td>927.00</td>
<td>927.00</td>
<td>435.00</td>
</tr>
</tbody>
</table>

Standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

B.2 Definition of high reciprocity types

In Table 4 of the paper, we classify developers into two categories – high reciprocity and low reciprocity – according to the median value of our field measure of reciprocity. This approach is consistent with the lab-in-the-field literature on reciprocity, which typically distinguishes between “weak” and “strong” reciprocators Fallucchi et al. (2017). Our results are relatively robust to variations in the way we define high reciprocity types. In table A5 we define a high reciprocity type as a developer that has a field measure of reciprocity in the top quartile. Then, instead of assigning discrete types to all developers, table A6 reports the impact of our field measure of reciprocity when averaged over all developers in the project.
### Table A5: High as 75 percentile

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>success score</td>
<td>project dead</td>
<td>success score</td>
</tr>
<tr>
<td>share of high reciprocators (top quartile)</td>
<td>-0.01</td>
<td>0.16***</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>(0.10)</td>
<td>(0.06)</td>
<td>(0.11)</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.22</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>N. of obs</td>
<td>927.00</td>
<td>927.00</td>
<td>435.00</td>
</tr>
</tbody>
</table>

Standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

### Table A6: Success and survival with average reciprocity

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>success score</td>
<td>project dead</td>
<td>success score</td>
</tr>
<tr>
<td>average reciprocity in project</td>
<td>-0.13***</td>
<td>0.07**</td>
<td>-0.04</td>
</tr>
<tr>
<td></td>
<td>(0.05)</td>
<td>(0.03)</td>
<td>(0.04)</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.22</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>N. of obs</td>
<td>927.00</td>
<td>927.00</td>
<td>435.00</td>
</tr>
</tbody>
</table>

Standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

### B.3 Proportion of team members with a field measure of reciprocity

In the main specification of table [4], we imposed the constraint that at least one third of the group members should have a measure of field reciprocity to guarantee that the average share of high reciprocators was not measured with excessive noise. We now relax and change this constraint. In Table [A7], we remove the constraint and in Table [A8], The result on the death of the project is preserved. The result on the effect of the share of high reciprocators on the success of the project when restricting projects to those still active (column (3) in the tables) is no longer significant,
although the magnitude and the size remain the same. For the case of no constraint, the standard
deviation increases because the measure is less precise. For the case of the stronger constraint, the
issue is that the number of observations is lower.

**Table A7: No constraint on proportion**

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>success score</strong></td>
<td>0.11</td>
<td>0.14***</td>
<td>0.15</td>
</tr>
<tr>
<td><strong>project dead</strong></td>
<td>(0.08)</td>
<td>(0.05)</td>
<td>(0.09)</td>
</tr>
<tr>
<td><strong>R-squared</strong></td>
<td>0.22</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td><strong>N. of obs</strong></td>
<td>1011.00</td>
<td>1011.00</td>
<td>473.00</td>
</tr>
</tbody>
</table>

Standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

**Table A8: At least half of project members with measure of reciprocity**

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>success score</strong></td>
<td>-0.01</td>
<td>0.23***</td>
<td>0.17</td>
</tr>
<tr>
<td><strong>project dead</strong></td>
<td>(0.09)</td>
<td>(0.05)</td>
<td>(0.11)</td>
</tr>
<tr>
<td><strong>R-squared</strong></td>
<td>0.20</td>
<td>0.12</td>
<td>0.13</td>
</tr>
<tr>
<td><strong>N. of obs</strong></td>
<td>806.00</td>
<td>806.00</td>
<td>392.00</td>
</tr>
</tbody>
</table>

Standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01

**B.4 Robustness of Table 6**

Table 6 addressed the question of whether a high share of reciprocators served as an amplification
of variations, making it both more likely to fail in bad times and more likely to succeed in good
times. To define good and bad times, we used the past history of the project and in the main
specification of table 6) use the three-month lag. We explore robustness by considering a two-month lag in Table A9 and a four-month lag in Table A10.

TABLE A9: TWO PERIODS LAG

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no contribution</td>
<td>total number of commits</td>
<td>total number of commits</td>
</tr>
<tr>
<td>no contribution last 3 periods</td>
<td>0.14***</td>
<td>-1.52**</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td>(0.72)</td>
<td></td>
</tr>
<tr>
<td>interaction with high reciprocity type</td>
<td>0.10***</td>
<td>-2.59***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td>(0.90)</td>
<td></td>
</tr>
<tr>
<td>high reciprocity type</td>
<td>-0.11***</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>above median contributions last 3 periods</td>
<td></td>
<td>2.27**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.96)</td>
<td></td>
</tr>
<tr>
<td>interaction with high reciprocity type</td>
<td></td>
<td>3.07**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.19)</td>
<td></td>
</tr>
<tr>
<td>R-squared</td>
<td>0.12</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>N. of obs</td>
<td>2.1e+05</td>
<td>2.1e+05</td>
<td>2.1e+05</td>
</tr>
</tbody>
</table>

Standard errors in parentheses
' * p<0.1, ** p<0.05, *** p<0.01
### Table A10: Four periods lag

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no contribution</td>
<td>total number of commits</td>
<td>total number of commits</td>
</tr>
<tr>
<td>no contribution last 3 periods</td>
<td>0.12***</td>
<td>-1.13**</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td>(0.56)</td>
<td></td>
</tr>
<tr>
<td>interaction with high reciprocity type</td>
<td>0.09***</td>
<td>-2.16***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td>(0.73)</td>
<td></td>
</tr>
<tr>
<td>high reciprocity type</td>
<td>-0.10***</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.01)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>above median contributions last 3 periods</td>
<td></td>
<td>2.32**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.10)</td>
<td></td>
</tr>
<tr>
<td>interaction with high reciprocity type</td>
<td></td>
<td>3.53**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.43)</td>
<td></td>
</tr>
</tbody>
</table>

| R-squared                     | 0.11                         | 0.01                         | 0.02                         |
| N. of obs                     | 2.1e+05                      | 2.1e+05                      | 2.1e+05                      |

Standard errors in parentheses

* p<0.1, ** p<0.05, *** p<0.01