What Make Valuable Contributors: Willingness and Opportunity in OSS Community

Minghui Zhou, *Member, ACM,* and Audris Mockus, *Member, IEEE*

Abstract—**Motivation:** To survive and succeed, software projects need valuable contributors (VCs). Only contributors with extensive project experience could accomplish critical tasks, yet few users become contributors and few contributors become VCs. **Aim:** We measure, understand, and predict how individuals capacity, willingness, and environment impact odds of becoming a VC. **Method:** Issue tracking data of Mozilla and Gnome, interviews, and online documents were used to estimate numbers of participants, and to design measures of capacity, willingness, and environment. A Logistic regression model was used to explain and predict contributor's odds of becoming a VC. **Results:** Nine measures of capacity, willingness, and environment were constructed. During their first month, future VCs were more active and community-oriented than other joiners. Newcomers who succeeded in having at least one reported issue to be fixed, more than double their odds of becoming a VC. The macro-climate with high project popularity and the micro-climate with low attention from peers reduced the odds. The precision of VC prediction was 72 times higher than for a random predictor. **Conclusions:** The findings provide a basis for empirical approaches to improve the retention of contributors and suggests more effective ways to contribute.

Index Terms—Valuable Contributor; Long Term Participant; open source; willingness; environment; interaction of person and environment

◆

1 INTRODUCTION

Successful open source software (OSS) projects need contributors because "OSS doesn't work without contributions from the community" according to an interviewee for this study. Important project tasks can only be accomplished by participants who have spent a substantial time with the project [1], thus retaining contributors over long periods is critical for success. Users are the prime candidates to become contributors [2], but only few users contribute and even fewer contribute over long periods. We found that only 0.02% of Mozilla users contributed by reporting or commenting on an issue and only 20% of the participants were active for three or more years (we refer to them as Long Term Participants or LTPs) as shown in Figure 1. Gnome had similar fractions: 0.1% of users contributing and 20% of participants becoming LTPs. The fraction of LTPs is rapidly dropping over time as illustrated in Figure 2. The strikingly low and rapidly dropping rate of LTPs motivated us to understand what happens to the newcomers in these two projects, and what factors are associated with participants staying with the project. Our aim is to use such understanding to help projects train and retain developers until they become Valuable Contributors (VCs) capable of solving critical tasks.

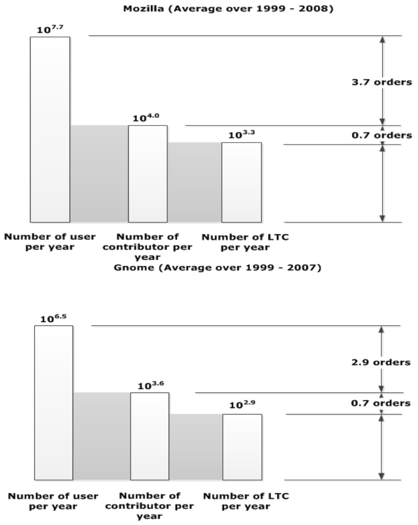

Fig. 1: Average number of users/new contributors/LTPs per year in Mozilla and Gnome

Roughly only one third of participants contribute more than once. We, therefore, focus on understanding of what happens at the time of the first contribution. Borrowing a framework from management science[3], which uses three interactive dimensions of capacity (e.g., ability), willingness (e.g., attitude), and opportunity (e.g., environment) to account for the work performance of individuals, we model the probability that a new joiner would become a VC, through her willingness and opportunity to contribute at the time of joining.

Issue tracking systems record the history of how people initiate and complete various tasks in software projects. We, therefore, assume that such detailed data

---

- *Minghui Zhou is with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, 100871.*
  E-mail: zhmh@pku.edu.cn
- *Audris Mockus is with Avaya Labs Research, 211 Mt Airy Rd, Basking Ridge, NJ.*
  E-mail: audris@avaya.com

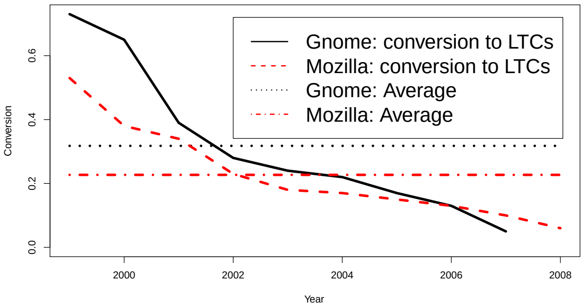

Fig. 2: Conversion of newcomers to LTPs over years

would contain traces that reflect people's ability, attitude, and environment and use issue tracking data to model willingness and opportunity in two OSS projects – Gnome and Mozilla.

We gathered and inspected artifacts recorded in the issue tracking system, in historic snapshots of projects' web pages, in published literature, and responses to our interview and survey. We used these sources to model a participant's willingness and capacity through the activities she is willing and competent to take on (the number and type of tasks she starts with) and the effort she is willing to provide (the value of her activities to the product and community, e.g., the fraction of reported issues that were ultimately fixed). We model opportunity through environment's macro-climate shared among all participants and micro-climate unique for a person. In particular, project's status such as popularity and relative sociality (RS)[1] constitute measures of macro-climate, while the initial size of peer group and actions of peers (e.g., their performance, their social clustering, or their attention to her) represent micro-climate.

We fit a logistic regression model[2] to quantify the relationships between these predictors and the chances of becoming a VC. A VC is operationalized as a productive LTP (above the 10-th percentile among LTPs based on the yearly activities).

The results show the probability to be associated with the contributor's attitude and environment. Specifically, at the time of joining, future VCs tend to take more active role and show more community-oriented attitude than other joiners. They also receive more attention from the community and encounter more experienced peers.

To evaluate if it is possible to use the results in practice, i.e., to tell who will stay in the project based on their first month of data, we conduct a prediction on 25406 newcomers of Mozilla. The prediction shows a precision of 24% that was 72 times higher than for a random predictor.

We also conducted a follow-up survey involving 40 participants (20 VCs and 20 one-time-contributors) to confirm that the measures of willingness and environment and the modeling results were consistent with participants' views.

The main contributions of this study include:

- The measures of participants' environment (micro- and macro-climate) and willingness (initial actions);
- An explanatory model of how the willingness and environment affect the odds of becoming a VC;
- A practical method to predict who will become a VC based on the initial actions and environment of a joiner;
- The quantification of trends reflecting users' conversion to participants and participants' conversion to LTPs.

The results suggest practical implications for practices to retain new contributors by, for example, devoting more attention to new contributors who report an issue. The newcomers who want to be welcomed by the community would benefit by focusing on the quality of their issue reports and on the community-oriented initial interactions.

To facilitate the reproduction of the study and of the proposed measures in other contexts we provide the data we retrieved and the scripts we wrote at *http://www.passion-lab.org/projects/developerfluency.html*.

We review related work in Section 2 and describe the project context and the methodology in Section 3. Section 4 presents our findings and Section 5 validates the results. We consider the limitations in Section 6, and conclude in Section 7.

2 RELATED WORK

Enormous effort over past decades was spent in attempts to unravel the possible relationships between job performance and its hypothesized antecedents in management science, cognitive science, and psychology. For example, it is widely believed that the characteristics responsible for exceptional performance are innate and are genetically transmitted [5]. However, Ericsson et al [6] showed that many characteristics once believed to reflect innate talent are actually the result of intense practice extended for a minimum of 10 years. Also, environment, for example, access to teachers, training material, and training facilities, rather than talent, are found to be the important factors determining the initial onset of training and ultimate performance [6]. Since Hawthorne studies [7], researchers have examined the effect on performance of formal and informal groupings, peer pressure, roles, norms, cohesiveness, goals, rewards, feedback, task characteristics, and other variables.

The most obvious conclusion from these varied approaches is that the variables known to influence individual task performance are numerous and varied. To summarize the existing theories, Blumberg and Pringle [3] proposed a three-dimensional-interaction

---

[1] A geometric average over all project's participants of the ratio of the number of individual's workflow peers (social aspect) to the number of tasks that individual participates in (technical aspect) [4].

[2] Logistic regression is a common way to model proportions where the proportion is related to predictors via a logistic function.

model of work performance, in which capacity, willingness, and opportunity are recognized as three interacting dimensions that account for the performance at the individual level. Capacity refers to the physiological and cognitive capabilities that enable an individual to perform a task effectively, including ability, skills, level of education, and endurance. The psychological and emotional characteristics that influence the degree to which an individual is inclined to perform a task comprise the willingness dimension including motivation, personality, and attitude. Opportunity consists of the particular configuration of the field of forces surrounding a person and her task that enables or constrains that person's task performance and that are beyond the person's direct control.

In this study we borrow this three-dimensional framework for one reason – ability, motivation/attitude and environment are the most popular concepts used by scholars in software engineering who have been at work attempting to predict developers' performance. For example, Curtis [8] claimed that the individual differences among project personnel account for the largest source of variation in project performance. Couger and Zawacki [9] identified how differences in the motivational structure of programmers interacted with the kinds of jobs they were assigned. They found that programmers had higher needs for personal growth and personal development than those in any other job category measured. Nakakoji et al. [10] studied four OSS communities and found the role that an OSS member plays in the community depends on how much the member wants to get involved in the whole community.

Environmental factors affecting developer's performance include organization variability [11], coworkers [12], and communication media [13]. Studies of newcomer experiences by Dagenias et al. [14] identified early experimentation, internalizing cultures, and progress validation as three primary factors facing developers joining new projects. Ducheneaut [15] found the successful participants in open source projects progressively construct identities as software craftsmen, and also progressively enroll a network of human and material allies to support their efforts. Earlier we have discovered [4] that the project's relative sociality when a developer joins impacts the probability she will become a long term contributor in that project.

In summary, ability, attitude, and environment comprise the three dimensions that, in literature, are often suggested to account for the developer's performance. However, the quantification of these relationships in software development contexts has been elusive.

3 METHODOLOGY

In this study we perform qualitative and quantitative analysis of two OSS ecosystems, Gnome and Mozilla[3]. A qualitative investigation is used to understand joiner behavior and to help design suitable measures of attitude and environment. Gnome and Mozilla have been extensively studied in the past. We, therefore, feel an obligation not to interfere with the work of project participants, to the extent we can rely on the results obtained in prior studies. Consequently, we chose to base our qualitative study primarily on recorded artifacts in the issue tracking system, project web pages, the existing literature, and the relevant websites. For questions where we could not obtain needed information from existing sources, we conducted a small survey. Most of information in open source projects is public and we, therefore, feel that such primarily record-based qualitative investigation would be able to capture the essential features of the investigated phenomena.

TABLE 1: Projects

Project	Years	MLOC [4]	Domain	Cntrbtrs
Gnome	10	7.9	UI	156,332
Evolution		0.8	Calendar&Mailbox	21,041
Nautilus		0.1	File manager	17,430
Epiphany		0.1	Browser	3,716
Mozilla	12	20.0	UI	187,333
Firefox		5.3	Browser	47,690
Thunderbird		1.1	Mailbox	12,993
Calendar		0.8	Calendar	4,130

Our quantitative study starts from Section 1, we compare the numbers of users, regular participants, and LTPs to demonstrate that only few users become participants and even fewer become VCs. Some LTPs stay with the project for a long time but contribute to only a few issues. We, therefore, exclude 10% of the LTPs with the lowest productivity (issues per/year) to obtain a subset of LTPs that we use to represent VCs in the later analysis. Then we focus on analyzing issue workflow to establish the relationship between contributor's probability of becoming a VC and the factors that measure her capacity, willingness, and environment.

We start from describing the project context in Section 3.1, introduce the qualitative study in Section 3.2 and the data retrieving in Section 3.3. We present our approach of counting participants (for Figure 1 and 2) in Section 3.4, and the workflow analysis in Section 3.5.

3.1 Context

Gnome and Mozilla implement user interface functionality, and have more than 10 years of history, as described in Table 1. Some major sub-projects in each ecosystem are also shown in the table. Evolution is the largest Gnome project, and Firefox is the largest of Mozilla's project. Note, that both ecosystems have a browser and a mail client.

3.2 Qualitative Study

For reasons noted above, we chose to do our qualitative study primarily based on digital records via the following procedure:

---

[3] Both contain a number of sub-projects

[4] data from ohloh.net

- We read the existing literature, particularly about Gnome and Mozilla, e.g., [16], [17], [18], to understand the project context and practices;
- We inspected the project web site looking for the project-related information, for example, the standard workflow of resolving issues. We also looked at the sub project web-pages, searched for relevant information, e.g., the practices used to report and resolve an issue;
- We sampled 40 people (20 non-VCs and 20 VCs) from each ecosystem, and carefully read the defects they were involved in, particularly at the time of joining, to understand the joining process and joiner experiences.

We also wanted to get a broader understanding of how people think of the factors influencing individual's performance in software projects. We, therefore, communicated with people (we know) from different companies including Google, Microsoft, Tengxun (a big Internet company in China), and Kingrain (a small software company with 13 developers in China). We asked the following question:

- What is the factor that you think has the most influence on the individual's performance?

After the response to the first question, we asked the second question related to factors not mentioned in the response:

- Do you think environment/ability/willingness affect individual's performance as well?

Finally, we sent a small survey to eight participants randomly selected from each project. In the email we listed the issues they reported and the following questions:

- What motivated you to report/comment on these issues? Did you report because of your personal interest or because of the business requirements you served at that time?
- Are these the only issues you have experienced? If not, why did you report only these ones?

We obtained one reply from each project, (five emails couldn't be delivered) giving us the response rate of 18%.

Once we completed the qualitative and quantitative studies and presented the results to (and eliciting feedback from) academic audiences, we conducted a follow up survey. We had two objectives:

- to validate our results;
- to observe if there have been changes over the two years since the initial study.

We sampled 40 people (10 VCs and 10 non-VCs from each ecosystem) and sent them the same questions as the earlier survey but adding an extra question asking non-VCs why they stopped contributing or asking VCs what made them continue contributing. We devoted attention to the individuals and tailored emails to reflect what each participant did. For example, "Your last contribution was commenting on issue XXX". Eight emails could not be delivered, and we received eight responses (the response rate of 32%).

3.3 Retrieving Issue Tracking Data

We obtained issue tracking data of Gnome and Mozilla. Traditional software projects use an Issue/MR (Modification Request) system to track defects, enhancements, and other project tasks[5]. The primary users are developers and testers. Customer issues are typically tracked in a separate system and only a very small subset that requires code changes, may be copied/imported into the system used by software developers. In contrast, in OSS software projects the issue tracking systems not only track tasks for developers and testers, but also track issues raised by end users and by down-stream projects. Each issue/MR has a history, from the time somebody reported it until the time somebody closed it (it also may remain open at the time of the study). During that period a sequence of events takes place: issue is created, assigned, resolved, tested, and closed. It may also be reassigned, its attributes changed, comments, debugging traces, etc added. Each such event has an associate date, time, the type of action, and the email/name of the actor.

Crawlers were written to obtain the issue histories and details from their web pages. Both Gnome and Mozilla use Bugzilla to track issues. We obtained information for all issues in XML format[6] as well as the activity history[7] for each project from all the sources in January, 2011 (each retrieve of the whole Bugzilla of Mozilla or Gnome is called a Bugzilla extract in this study). There are not many issues prior to 1998 in Mozilla and very few prior to 1999 in Gnome, hence we removed data before 1998 in Mozilla and before 1999 in Gnome. Overall 158,244 user ids and 517,801 MRs were in Gnome, and 200,655 user ids and 620,511 MRs were in Mozilla.

After the completion of the study and the presentation of our results, we retrieved issues for Mozilla again in May 2012. We used this extract to validate model predictions on a completely new data. This later extract had 214,576 user ids and 709,386 MRs.

3.4 Estimating Numbers of Users and Participants

Even though Bugzilla identifies participants via email, name, or login, the record is not perfect and it may change over time. This may lead to the same identification being associated with multiple individuals (multi-person ID) or to several distinct identifications for the same individual (multi-ID person). We, therefore used five Bugzilla extracts obtained by different individuals at different times, as shown in Table 2.

From each extract we gathered information for all participants: email (or, if not available, login[8]), name

---

[5] http://en.wikipedia.org/wiki/Issue_tracking_system

[6] e.g., https://bugzilla.mozilla.org/show_bug.cgi?ctype=xml&id=3549

[7] e.g., https://bugzilla.mozilla.org/show_activity.cgi?id=3549

[8] Because Gnome Bugzilla prevents retrieval of large numbers of issues with complete email address of participants, Gnome 2011 extract has only their logins instead of emails.

TABLE 2: Five Bugzilla Extracts

Name	Author	URL
Gnome 2011	authors	passion-lab.org/projects/developerfluency.html
Mozilla 2011	authors	same
Mozilla 2012	authors	same
Gnome 2006	P. Wagstrom	academic.patrick.wagstrom.net/research/gnome
Mozilla 2008	C. Birds	msr.uwaterloo.ca/msr2009/ challenge/msrchallengedata.html

(if available), joining date (the date of the participant's first activity), leaving date (the date the participant's last activity), and whether or not the participant was an LTP (i.e., the leaving date was at least three years after the joining date).

Bugzilla records do not provide sufficient information to identify individuals without uncertainty. We, therefore, compared several approximate identifications that are based on email, login, and full name. Once individuals were identified, we counted the number of new participants and the number of future LTPs joining each year. Notice that we can identify joining LTPs only for dates that are more than three years prior to the date of the extract.

3.4.0.1 Email- and login-based identification.: A single participant may have multiple emails and/or logins, e.g., 10% of the full names are associated with at least two different emails in Mozilla 2011 extract. A participant who changes her email after joining the project would be considered as several different participants if email is used as a proxy for a person.

On the other hand, a single email or login may be used by multiple participants. For example, it is common for certain roles, such as product administrators, to share the same email.

Furthermore, the same login may represent different emails, possibly of different participants. For example, emails zhmh@pku.edu.cn and zhmh@avaya.com share the same login zhmh, but they may not be representing the same person. In particular, the 2011 extract of Mozilla has 184379 unique emails, but only 154351 unique logins. It's possible that there may be two different non-LTPs with the same login and participating three years apart who will be identified as a single LTP.

3.4.0.2 Name-based identification.: A participant may be identified by the associated name and all emails/logins used for a particular name may be linked to a single person. This approach also suffers from the issues identified above: the same name may be used by multiple people or the same person may change their name. Furthermore, not everybody provides their name, so for some participants only email (or login) is available.

To assess how sensitive our results are to the person identification method we selected names that had more than one identification associated with them (i.e., multi-ID names) and identifications that had more than one name associated with them (i.e., multi-name IDs). We then obtained the counts of the participants and LTPs by removing multi-name IDs and by replacing IDs for each multi-ID name by the corresponding name. Using this approach the counts were similar for different extracts with IDs being emails or logins. However, the counts of LTPs obtained using email were substantially smaller than the counts of LTPs obtained using just the login. Investigation of the discrepancy suggested that a sizable number of short term participants pairs who did not provide their names happened to have matching logins and were misclassified as LTPs. Furthermore, the most recent Gnome extract (2011 extract) did not contain email addresses and the remaining two Gnome extracts cover relatively short time periods, e.g., LTPs could be obtained only before September 2005 using Gnome 2008 extract. We, therefore, could not use emails as IDs.

After evaluating several approaches we settled on reporting the numbers obtained by using name as an identifier of the individual. Indeed the approach of using names has its advantages. On one hand, this approach applies for all extracts, and the numbers obtained from different extracts of the same project were quite similar. On the other hand, people tend not to change their names as often as they do with emails and logins. Some participants do not provide their names,e.g., in Mozilla 2011 extract, 19% of the 184379 emails do not have an associated name[9].

In summary, the numbers of participants reported in Figure 1 and 2 are obtained based on the names and include only individuals who provided their names. To model the longest time period we report results obtained from the Gnome 2011 and Mozilla 2012 extracts. It is worth noting that the ratio of newcomers to LTPs is larger when obtained by using the names as IDs in comparison to using emails or logins as IDs. It suggests that the contributors who are willing to provide their names are also more likely to become LTPs and is consistent with the findings in Section 4.1 on the effects of the higher values for measures of willingness.

We obtained the size of user population of Gnome and Mozilla from various sources available online, such as Wikipedia, browser market share counters, and annual reports of Gnome foundation. To approximate the number of Mozilla users we multiplied the market share of Firefox by the estimate of the total number of Internet users. Similarly, we multiplied the market share of Ubuntu/Gnome platform. More details are provided in Section 4.2 and Section 4.3.

3.5 Issue Workflow Analysis

We investigate the workflow issue resolution to measure people's capacity, willingness and opportunity. The issue workflow in Bugzilla records traces of participants' activities while resolving issues and we use it as the basis to construct measures reflecting the participants

---

[9] This is one of the reasons why we didn't use names to represent participants for later study, instead, we used emails for Mozilla, and logins for Gnome, see Section 6 for more detail.

behavior and project environment. In the following we introduce the workflow details necessary for later analysis in Gnome and Mozilla.

OSS projects often define a protocol to follow when reporting, confirming, and resolving issues. This is often referred to as a triage process. In particular, Gnome defines the standard steps of triaging on its website[10]. According to it, an issue is reported (born) in an UNCONFIRMED state. The issue may be confirmed by changing its state to NEW or, it may be immediately resolved and the state is changed to RESOLVED. When additional information from the issue reporter is needed to proceed further in fixing this issue, the state would be changed to NEEDINFO. The transition from one state to another may involve a change of actors. We consider each such transition as an artifact-mediated communication between two of the adjacent actors. Newcomers take different actions based on their capacity and willingness, as described in later sections. Notice, not everybody is allowed to modify the state of an issue. We have verified that a new contributor is allowed to modify the state only for the issues she reports, but she is able to comment on any issue. Mozilla's triage process is similar[11]. However, in practice the process may be different from the protocol described in the guidelines. For example, many issues started from NEW instead of UNCONFIRMED, as Table3 shows. An investigation[19] suggests that one of the reasons involves core developers reporting issues in a NEW instead of UNCONFIRMED state.

TABLE 3: The starting status of issues

Status	Gnome	Mozilla
UNCONFIRMED	0.84	0.53
NEW	0.15	0.46

Each "RESOLVED" issue has a resolution, e.g., FIXED, DUPLICATE, INCOMPLETE, INVALID, or WONTFIX. The resolution types vary between Gnome and Mozilla, e.g., Mozilla resolution EXPIRED is used in similar cases as OBSOLETE in Gnome. As the name suggests, FIXED means the bug is fixed, DUPILICATE means the reported bug is a duplicate of some other bug, INCOMPLETE means the reported information is not sufficient to reproduce the bug, INVALID means this is an invalid report, and WONTFIX means that the reported issue is not relevant enough to be fixed. Table 4 shows the proportions of resolved issues for each resolution. In particular, FIXED issues represent 32% of Gnomes and 36% of Mozillas resolved issues.

TABLE 4: Ratios of issue resolution types

Status	Gnome	Mozilla
FIXED	0.32	0.37
DUPLICATE	0.36	0.24
INCOMPLETE	0.15	0.06
EXPIRED	0.03	0.03
INVALID	0.02	0.10
WONTFIX	0.03	0.05

4 RESULTS

Contributors interact with their environment when they join the project. That interaction is mediated by their capacity and willingness and it affects the odds of them becoming VCs. We start from quantifying contributor's capacity and willingness in Section 4.1. We continue with measuring contributors' environment in Section 4.2, and fitting a logistic regression model of the probability that a newcomer will become a VC in Section 4.3. At last we predict which new joiners will become VCs in Section 4.4.

4.1 Measuring Capacity and Willingness

Software development is a knowledge intensive activity [20], and the almost universal assumption of personnel managers is that personality has a marked effect on the performance of employees [3]. As an employee from Google commented in our interviews, *"basically, personality determines everything"*. And *"environment is similar for everyone (here), attitude is a part of personality"*. How about individual ability? *"We are Google, never lack talented guys"*. However, another interviewee (a manager from Kingrain) said *"ability accounts for everything"*, because his best employee is substantially more competent than the remaining employees. As Blumberg and Pringle [3] clarified, personality, attitude, and motivation are variables of "willingness to perform", while ability, experience, and intelligence are variables of "capacity to perform". In this section we measure the capacity and willingness of newcomers to the OSS project, and try to establish if the contributor's willingness measured at the time they join the project varies between participants who will and who will not become VCs.

Only 3.6% of Gnome and 0.9% of Mozilla joiners become VCs. And in both projects, more than 70% of contributors are one-time-contributors (OTCs, 70% in Mozilla, 78% in Gnome). These contributors have only a single interaction (e.g., reporting an issue or commenting on an issue) recorded in the issue tracking system. This high level of peripheral participation is consistent with the finding that more than three-quarters of the nearly 13 thousand contributors Lerner and Tirole [21] considered, made only one contribution. We sampled 20 OTC and 20 VC participants from each project. For ethical reasons, we omitted the participants' names and represent them using numeric aliases, with 1-20 representing OTCs and 21-40 representing VCs.

We manually inspected all the issues reported by OTCs, and observed two types of behavior. Both of these behaviors were exhibited by contributors who appear to be end users. Some were eager contributors who worked thoroughly and tried to help. For example, gnome-7 reported: *"I've been trying to use libxml++ ... However I*

---

[10] http://live.gnome.org/Bugsquad/TriageGuide

[11] https://bugzilla.mozilla.org/page.cgi?id=fields.html

found that there are two rather large memory leaks". He even committed some attachments to help fix, *"Please let me know if these issues will be addressed"*. Some people proposed specific requirements. For example, mozilla-15 asked to implement a new feature, *"I'm writing to request support for JIF be added to mozilla"*. For this he was willing to help – *"I can provide some help for this, but I would like to know if the mozilla developers are receptive to this"*. Another group appeared to be not devoting as much attention to the issues they reported. For example, gnome-12 reported two issues according to the required format but without any extra comments. Furthermore, he/she didn't respond to request for additional information: *"Could you please help fixing this by installing some debugging packages..."* (the issue was resolved with the state INCOMPLETE). These are some of the reasons to believe that these people reported issues (and committed fixes) because they were end users and they were trying to accomplish something relevant with the OSS product they were using. At the same time, they might have found the issues by accident (because they were users), and they were interested enough to report it but were not motivated enough or had no opportunity to contribute more.

For the sampled VC group, the average number of MRs per person was over 100 (over the considered period, see Section 3.5). We, therefore, examined only a subset of their issues, in particular, issues in their first month after joining. Notably, they all were more active as compared to the OTC group: made more comments, and often spent more effort on their issue reports, for example, by including a patch. For example, mozilla-25 reported his first issue on Oct 24, 2007. Based on the comment on that issue: *"Reporter, could you reproduce on FF3 RC?"*, we assume that he was not known by the community at that time. Less than one month later, on Nov 21, mozilla-25 created attachments for a couple of issues. Apparently, he already won trust from the existing developers by, perhaps, submitting good quality patches, because a developer applied to obtain an SVN account for mozilla-25 the same day (through Bug x: Create localizer LDAP/SVN account for mozilla-25). A similar example in Gnome is gnome-30, who reported her first issue on Mar 20, 2004. This report was committed with a couple of extra attachments showing the details needed to understand and reproduce the issue, and the issue resolution was "FIXED". Another example is gnome-33, whose first report was *"Patch to get access attributes for nested class/struct/union"*. In other words, the first action he did is to commit a patch in the form of an issue report, and it appears to have been useful, because the responsible developer responded – *"I'll include it in the first CVS release"*.

In summary, all OTCs and most VCs appear to be product users at the time of joining. The two survey responses were *"I use a lot of Open Source software both at work and at home (including Mozilla)"*, and *"Around that time, I did the final move away from Windows and replaced Windows on my desktop PC at home (with Gnome)"*. This observation is consistent with the findings in project Fedora: *"74% said they were first users of Fedora and then became contributors to the project"*[12]. Moreover, both groups show ability and willingness to contribute — out of millions of Mozilla/Gnome users, only a very small proportion is going to contribute, no matter if the contributions are "in the form of bug reports, suggestions or occasionally code contributions". Also, quoting from our survey: *"I did report the issues because of my personal interest"*, *"as a technically knowledgeable user I feel a responsibility to give back when I am able"*.

However, what are the differences among contributors, in particular, between OTCs and VCs? The evidence appears to favor willingness, i.e., the extent to which people are getting involved in helping the project.

We recognized three aspects of measuring the extent of participation. First we consider the number of tasks, e.g., number of comments she makes. These comments might be an interpretation for a confusing report, or suggesting a possible solution, or explaining the benefits of a proposed new feature. We assume that the number of tasks primarily reflects the skills the participant has accumulated prior to joining the project, i.e., her capacity, but also, partly, how willing she is to contribute.

The two other aspects include what type of tasks a participant takes on and how much effort she provides. The activities a participant undertakes, e.g., attaching the screenshot when reporting an issue is a sign that she has a community-friendly attitude. Starting from a comment instead of reporting an issue also reflects such an attitude — it means she is intentionally getting involved, perhaps by first finding a similar issue and commenting on it instead of simply reporting an issue she encounters as a user. On the contrary, reporting an issue through a crash-reporting tool such as Bug-Buddy is extremely easy and does not demonstrate a great amount of desire to get involved. When an application using the Gnome libraries crashes, Bug-Buddy generates a stack trace and invites the user to submit the report. This requires little effort from the user, simply filling a few fields and clicking "Submit" button. In contrast, alternative way of reporting an issue involves applying for an account for Gnome Bugzilla, creating a new issue report, and filling in the template that includes steps needed to reproduce the bug. In Gnome, less than 1% of the joiners, who had their first issue reported via Bug-Buddy, eventually became VCs, while more than 4% of the joiners who started with a Bugzilla report became VCs.

Even though contributions appear to be universally valued in open source projects, the following statistics casts doubt on that assumption, or, at least, suggests that some contributions provide much more value than others. Approximately 90% of the OTC reports ended

---

[12] http://www.cyber-anthro.com/beta-an-exploration-of-fedoras-online-open-source-development-community/

TABLE 5: Quartiles of number of comments

Resolution	Gnome				Mozilla			
	25%	50%	75%	100%	25%	50%	75%	100%
DUPLICATE	1	1	1	283	2	3	5	722
INCOMPLETE	1	2	2	200	2	3	5	168
INVALID	1	2	4	247	2	3	6	384
WONTFIX	1	2	5	819	2	5	9	760
FIXED	2	3	6	1578	3	7	14	756

up without any change (i.e., without resolution FIXED, 91% in Mozilla, and 90% in Gnome), of which a big proportion were duplicates (47% in Mozilla and 35% in Gnome). However, for the defects reported by VCs, a high proportion were FIXED (51% in Mozilla, 52% in Gnome), and approximately 15% in both projects were duplicates.

This analysis shows how the issue resolution types can be used to measure the value a contributor provides and to reveal her attitude. Specifically, fixed issues help improve the product quality and might enhance community morale by showing some accomplishment. Meanwhile, the remaining issues might waste the limited time of the few core developers. Table 5 shows the differences between issues closed with resolution FIXED and issues with other resolutions. The issues resolved with FIXED tend to have more comments than other issues, and the issues with resolution DUPLICATE tend to have the least comments. It suggests that more effort and attention is devoted to fixed issues.

On the other hand, when a contributor encounters an issue, if she spends more time to search for similar issues and the relevant resolutions, the more likely she is to find exactly the same or similar issue. There is less chance, therefore, that the issue will end up with DUPLICATE resolution. The same argument applies for issues with resolution INCOMPLETE — responding and providing enough information for others to reproduce the issue would make it more likely that the issue will be fixed. In other words, if a person puts enough effort, the issues she reports have more chances to be fixed and, thus, improve the quality of the product. Consequently, the fraction of reported issues that are fixed (in the first month from joining) is an indication of effort a contributor provides.

Therefore:

Observation 1: A contributor's willingness (and capacity), can be measured by the number and types of tasks (e.g., reporting or commenting on an issue) she participates in and by the effort she provides to resolve these tasks.

4.2 Measuring Environment

Early literature on job performance predictions did not consider the environment: "since performance is ultimately an individual phenomenon, environmental variables influence performance primarily through their effect on the individual determinants of performance – ability and/or motivation" [22].

However, available evidence indicates that certain environmental factors beyond the individual's control play a far stronger role influencing her performance than was generally acknowledged then [3]. The more important of these involve what is known in normative decision theory as states of nature and actions of others, and suggest a clear recognition that, in addition to social, psychological, and physiological determinants, behavior (performance) also depends on the help or hindrance of uncontrollable events and actors in one's environment.

In our case we separate two aspects of the environment: macro-climate and micro-climate to account for states of nature and actions of others. Macro-climate represents the overall project environment that is the same for everybody in the project. Micro-climate represents the conditions that each individual encounters and varies among participants.

We start from measuring macro-climate. Based on the literature and our experiences, product popularity, project task density and project sociality are important elements. The market value of a product comes from its usage. In other words, the number of users ($users$) reflects the product's market proceeds, that will affect the funding (tools, equipment, materials, supplies, and pay) for the project and the degree of interest people would devote to it. Consequently, it will likely affect the contributors' stay with the project. We measure project popularity via its user population each month. For example, Firefox is the primary product in Mozilla and we use its user base as proxy for Mozilla user base. We obtained the user population by multiplying the market share of Firefox[13] by the total number of Internet users[14]. Project task density describes how much work is done in the project, and we measure it by the number of active MRs each month ($numMR$). This aspect of the macro-climate indicates whether the project is active and whether the participants have high workloads. Project sociality represents project's social climate and we could measure it by the participation density, i.e., number of participants each month ($newJoiner$), or relative sociality (RS)[3]. Figure 3 shows the evolution of macro-climate components in Mozilla. We were able to obtain Internet user estimates starting from Dec, 2000. Figure 3 also marks the calendar times of important releases of Firefox, e.g., in Jan, 2010, Firefox 3.6 was released. We have divided the user numbers for each month by the highest number of users over the entire period. Thus, this variable ranges in value between zero and one. Notice, in order to present the curves on a similar scale, we normalized each, e.g., by dividing $newJoiner$ by 50, as shown in the legend of the chart. The project's popularity grows from the start of the project, because Internet user population grew and Firefox share has increased. The task density is high when the project is close to its release date, and participant density is high when the task density is high.

These components of macro-climate are the same for

---

13. http://en.wikipedia.org/wiki/Usage_share_of_web_browsers
14. http://www.internetworldstats.com/stats.htm

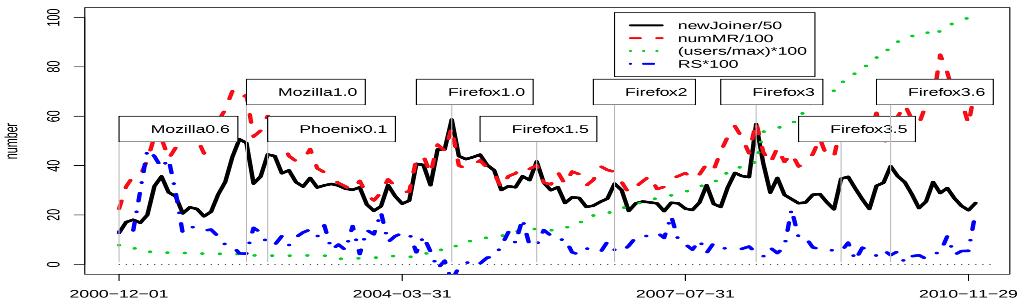

Fig. 3: Macro-climate in Mozilla

all project participants at a particular time. Person's micro-climate may be characterized with the following Chinese proverb: "He who stays near vermilion gets stained red, and he who stays near ink gets stained black". We, therefore, consider people in her workflow network, i.e., people she interacts with, to create a unique environment for her. In other words, the actions or performance of these people, and her relationship with them, constitute the micro-climate for this person. We chose the initial size of a participant's social group, their productivity, their social clustering, and the attention they give to her as measures of the micro-climate.

The size of a person's workflow peer group, i.e., the number of peers in her workflow network, is primarily determined by her own actions. The more issues she is involved in, the more likely she will meet more people. But the actual act of contribution is associated with the person's accumulated ability and willingness to contribute. At the same time, her peers constitute her social working conditions. According to [23], we assume that if a person has more peers, she is more likely to attach to the project, therefore she is more likely to become a VC. On the other hand, the performance of her peers is likely to affect her performance, as in an old saying: "a cat gets stronger with a tiger." For example, Mockus [24] found that more productive mentors lead to more productive followers.

We consider the social clustering to be the amount of replication among the workflow networks of peers. For example, Contributor Alice has two peers, Dragon and Tiger, and Dragon meets Lion and Bear, Tiger meets Lion and Deer, the sum of Alice's two peers' network sizes is $3 + 3 = 6$, but the size of the joint network is 4 (because Lion and Alice are repeated twice), therefore her social clustering is $\frac{6-4}{4}$. The underlying assumption is that if a person's peers have more in common (share more colleagues), it is more likely that they have similar project experiences and share similar values. The new participant, hence, is less likely to get confused by a variety of behaviors and value systems she observes. Furthermore, more clustered peer group is more likely to understand and trust each other, and that, in turn, might create a better environment for a newcomer to learn and to become more effective. It might also increase her work satisfaction and the willingness to stay. For example, Mayer [23] found the willingness to trust others was significantly related to the behavior and performance of individuals.

Humans need attention from other people, and developers are no exception, notwithstanding common stereotypes. Perhaps, the more attention a newcomer could obtain from the existing project members, the more likely she will stay with the project for a long time. We measure the amount of attention through the duration of time between the newcomer's first action until somebody responds. The response delay that is too short may not bode well in terms of attention. A very short response may imply that the responder did not take the issue seriously or did not inspect it carefully but just replied with a canned template to save time, e.g., *"Thanks for taking the time to report this bug. This bug report isn't very useful because it doesn't describe the bug well."* In these circumstances the reporter might feel under-appreciated and stop contributing.

Therefore:

Observation 2: A contributor's environments may be measured via macro- and micro-climate. Macro-climate represents environment shared among participants, and consists of project popularity, project task density and project sociality. Micro-climate represents environment unique for each person and consists of the initial size of peer group, their productivity, their social clustering, and the attention they devote to this new participant.

4.3 The Chances of an Individual's Success

We investigate the influence of the ability, willingness, and environment on the chances of an individual's success in the project by fitting a logistic regression model specified in Equation 1 with 125,665 observations in Gnome and 130,471 observations in Mozilla. The response is the indicator of a new participant becoming a VC and the predictors include measures of her ability, willingness, and environment described above (Notice not all attributes are presented because of the correlation, e.g., project task density is heavily correlated with RS). Each observation represents one project participant, with the predictors calculated over her first month from joining shown in Table 6. The predictors that require more explanation are discussed below.

We operationalized the size of peer group ($nPeer$) in two ways. The first approach counts the number of other participants she encounters in her first month's workflow. Second approach considers the number of participants encountered by her peers. Both measures have a similar association with the response, but we present the second measure because it explains more variance in the response and is not correlated with other predictors in both projects.

Popularity ($nUsr$) represents the number of Firefox users in Mozilla, as described in Section 4.2. We did the following to approximate the historic numbers of Gnome users. First we obtained the estimates of the fraction of linux users[15] [16]. Then, we used surveys of desktop choices for the period between 2003 and 2008[17]. For the 2009 to 2011 we approximated Gnome users by the fraction of Ubuntu users[16]. Eventually we multiplied the market share of Ubuntu/Gnome by the estimates of Internet users[14] to approximate Gnome user numbers.

We used $GotFix$ to represent the value contributor provided to the community. Having at least one of the reports to be fixed indicates a tangible improvement of product quality.

Barrier to entry BtE depends on the project. We used $FNotRep$ (the first participation is not an issue report) for Mozilla and $withBB$ (the first participation is using Bug-Buddy) for Gnome, because $withBB$ explains more deviance than $FNotRep$ in Gnome. Since Mozilla did not have an equivalent tool that required minimal effort to report an issue we used $FNotRep$ as a proxy for high willingness to contribute.

We used $LckAttn$ to represent an extreme situation of a too rapid response (within one hour).

Predictor prj is a sub-project indicator (not shown in the table) of the ecosystem the participant starts with, e.g., Evolution of Gnome, Firefox of Mozilla. It explains $1-2\%$ of the deviance and was added to account for the variation among sub-project environments.

$$\begin{aligned} isVC \sim nUsr + RS + GotFix + BtE \\ +nCmt + nPeer + pShared \\ +LckAttn + PeerPerf + prj \end{aligned} \tag{1}$$

Tables 7 and 8 contain fitted values for Gnome and Mozilla. 23% of the deviance is explained in Gnome model and 19% in Mozilla. The second column has the estimated coefficients, and the third standard errors. All predictors are significant (at < 0.01 level), except for RS in Mozilla. The forth to sixth columns show practical importance of the predictor in determining the VC probability through effect sizes. In logistic regression the effect size is odds ratio (i.e., the sixth column) for the mean value of the predictor (i.e., column labeled by x) and for the mean value plus the standard deviation (i.e., column labeled by x_{alt}). That's the case for predictors $nUsr$, RS, $pShared$, and $PeerPerf$. For predictors with low discrete values $nPeer$ and $nCmt$ we chose to use median and 75th or 90th percentiles to make the interpretation of the effect size more meaningful. For the boolean predictors such as $GotFix$, $withBB$, $FNotRep$, and $LckAttn$ the effect size is the odds ratio for the most frequent and the less frequent values. Two hypothetical Mozilla contributors: Alice with one comment in her first month and Bob with four comments (90th percentile of $nCmt$ is $\ln(4+1)$ or four comments), therefore the odds of Bob becoming an VC are 112% higher than odds for Alice if their remaining predictors have values specified in the forth column of the table.

TABLE 6: Predictors for participant i

Dimension	Predictor	Description
Willingness (and capacity)	nCmt	Logarithm of the number of comments +1
	GotFix	At least one of the issues reported by i was fixed
	withBB	First report by i uses a crash reporting tool
	FNotRep	i starts participation with a comment
Macro-climate	nUsr	Number of product users when i joins
	RS	Project's relative sociality
Micro-climate	nPeer	Logarithm of i's peers' group size: $\ln \Vert \cup_{p \in Peers(i)} Peers(p) \Vert + 1$, where $Peers(p)$ is the peer group for p
	pShared	Logarithm of the social clustering of i's peer group $\ln \frac{\sum_{p \in Peers(i)} \Vert Peers(p) \Vert}{\Vert \cup_{p \in Peers(i)} Peers(p) \Vert}$
	PeerPerf	Logarithm of the minimum productivity (issues/month) of the peers $\ln min_{p \in Peers(i)} nmr_p + 1$
	LckAttn	The longest duration between the i's action until the response is less than 1 hour

TABLE 7: Model for Gnome (125,665 observations)

	Est	Std.Err.	x	x_{alt}	$\frac{Odds(x_{alt})}{Odds(x)}$
(Intcpt)	-4.79	0.193			
nUsr	-1.95	0.0908	0.528	0.87	51/100
RS	-0.981	0.0588	-0.794	-0.468	73/100
GotFix	0.829	0.0354	F	T	229/100
withBB	-1.08	0.0556	T	F	295/100
nCmt	0.719	0.0314	ln 2	ln 4	165/100
nPeer	-0.0543	0.00673	ln 1779	ln 5405	94/100
pShared	2.00	0.182	ln 1.06	ln 1.17	122/100
LckAttn	-0.501	0.0778	F	T	61/100
PeerPerf	0.218	0.00496	ln 15	ln 318	195/100

In summary, the probability of becoming a VC is associated with the contributor's willingness and environment. Specifically, starting from comments instead of reports, reporting with Bugzilla instead of Bug-Buddy, or reporting any sensible issue that gets fixed double the odds of becoming a VC. Regarding micro-climate, low attention in the form of too rapid response would reduce the odds by 28% in Mozilla and by 39% in Gnome. Increase of the productivity of the slowest peer from 14 to 317 MRs/month in Gnome and from 14 to 248 MRs/month in Mozilla would increase the odds by 95% and 20%. Increasing the social clustering by

---

15. http://en.wikipedia.org/wiki/Usage_share_of_operating_systems
16. http://stats.wikimedia.org/archive/squid_reports/
17. http://www.desktoplinux.com/news/NS8454912761.html, http://www.desktoplinux.com/articles/AT2127420238.html

TABLE 8: Model for Mozilla (130,471 observations)

	Est	Std.Err.	x	x_{alt}	$\frac{Odds(x_{alt})}{Odds(x)}$
(Intcpt)	-7.49	0.419			
nUsr	-0.601	0.15	0.308	0.57	85/100
RS	0.701	0.293	0.0738	0.173	107/100
GotFix	0.74	0.0831	F	T	210/100
FNotRep	0.507	0.0821	F	T	166/100
nCmt	0.819	0.0409	ln 2	ln 5	212/100
nPeer	0.142	0.0205	ln 1650	ln 4266	114/100
pShared	2.35	0.135	ln 1.12	ln 1.32	148/100
LckAttn	-0.325	0.124	F	T	72/100
PeerPerf	0.0649	0.0131	ln 15	ln 249	120/100

0.11 in Gnome and by 0.2 in Mozilla leads to 22.15% and 48% increase in the odds. For the macro-climate, product popularity is associated with lower odds that a contributor becomes a VC — increasing the number of users by 34% in Gnome and by 26% in Mozilla reduces odds by 49% and by 15% respectively. Project's RS is associated with lower odds of becoming a VC in Gnome (0.326 of RS increase leads to 27% decrease in the odds) but higher odds in Mozilla (0.0992 of RS increase leads to 7% increase in the odds). Also, having the size of the peer group increasing from the median to third quartile is associated with a small decrease in the odds (6%) in Gnome and an increase (14%) in Mozilla. In summary:

Observation 3: The probability of a newcomer becoming a VC is associated with her willingness and environment. Her pro-community attitude that determines her choice to start with a comment instead of a report or to report with Bugzilla instead of automatic tool, and the amount of effort she provides to the community, are associated by the most dramatic increases. On the contrary, her macro-climate with high project popularity, and her micro-climate with low attention, reduce her odds. Meanwhile, the attributes of her peer group, in particular, its social clustering and productivity significantly influence her opportunity to become a VC.

4.4 Predicting Who Will Become A VC

To evaluate if it is possible to use the results in practice, i.e., to tell who will stay in the project based on the data collected during their first month of participation, we predict VCs among the 25406 Mozilla newcomers joining from January of 2008 until April of 2009.

We choose this group of people for the following reasons. They joined Mozilla prior to January of 2011, thus the data collected during their first month of participation was available to make the prediction using the model shown in Table 8. We have selected the 1% (254) of these joiners, who had the highest predicted probability to become a VC, as predicted-VCs.

However, it was not possible to tell which one will become a VC. A VC needs at least three years of participation and, therefore, we had to wait until April of 2012 to determine which of these participants would become VCs. To evaluate these predictions we retrieved another Bugzilla extract in May of 2012. From this extract we determined the 131 participants in the newcomer sample who became VCs (true-VCs). The fraction of predicted-VCs who were true-VCs was 24% (precision) and the fraction of true-VCs that were in the predicted-VC set was 47% (recall). For comparison, a predictor that randomly selects one percent of newcomers would have the precision of 0.516% or 72 times lower, and the recall of 1% or 47 times lower.

If we use the model to predict who will become VCs among the participants joining Mozilla from January of 1998 until December of 2007, i.e., the participants used to fit the model, the precision would be 22.15%, and the recall 24.6%. The prediction performance on the new dataset is even better than on the dataset used to fit the model, suggesting that the model can not capture the earlier history as precisely as the more recent trends.

In general, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. For example, for the prediction of VCs joining from January of 2008 until April of 2009, an increased precision (from 24% to 37.2%) is accompanied with a reduced recall (from 47% to 24.6%).

TABLE 9: Survey responses

	Delivered		
	Answered	Not-answered	Not-Delivered
Non-VCs	1(2)	12(9)	7(5)
VCs	7	12	1

5 VALIDATION

As noted in Section 3.2, we tried not to interfere with the work of project participants and, therefore, we surveyed only sixteen participants. Such a small sample, with five undeliverable addresses and only two respondents, did not appear to be sufficient. We, therefore, conducted another survey to increase the confidence in the findings from the survey.

We sampled 10 VCs and 10 non-VCs from each project, and tailored emails to each individual. In particular, we noted their personal contributions, as described in Section 3.2. Overall the survey responses were consistent with our prior findings, in particular, we obtained a clear support for the measures of willingness and environment that distinguish VCs from non-VCs.

First, the response rate suggests that VCs tend to have an attitude that is more community-oriented. As shown in Table 9, VCs had higher response rate than non-VCs in the survey, Fisher-test has a p-value of 0.07.

Second, participants' responses provide evidence on how these individuals' initial capacity, willingness and environment affected their decision of staying with the project or leaving.

We obtained only one response from non-VCs, yet it clearly states the reason for stopping contributions: *"I don't become VC because I don't have enough time/knowledge to resolve issues in Mozilla by myself (it is too complex for me)."* Also, the response articulates key difficulties of contributing: *"If you have faced a bug, you need to make some effort to describe it. Then you must check if there is duplicates. Then you create report and wait until response. All time you are waiting you must keep an issue in mind. After initial response there is good possibility that devs (developers) can't or don't want to reproduce the issue and you must know how to to diagnostics and how to prof that issue is really exists. Then you wait until issue is fixed usually without any feedback on progress. When issue gets fixed you should confirm this, but you waited for a long time and probably forgot some details. Also where to get fixed binaries? They were released or you must compile them from sources for your platform?"* Therefore, according to his experience, *"if I don't annoy developers constantly with the issue or don't fix it by myself it wont be fixed"*[18]. Eventually, the result is, *"To effectively contribute one must be professional developer with all the tools and knowledge at hand. Most people will give up or decide not to contribute at all after having some first negative experience."* This strongly supports our measure of *GotFix*, which measures the ability and willingness of the new participant, and has a significant impact on the odds of becoming a VC.

On the other hand, all VCs appear to be idealistic and motivated, for example, *"A single ant cannot do much, but many ants can move mountains. That is the strength of Open Source. I want to help move a mountain!"*, *"It is kind of like making the world a better place in small steps"*, and, *"They (contributors) want to make things better for everyone, not just themselves"*.

They are not only motivated to make the world better, but also aware how they are going to do that – they pick the tasks they are capable of doing. For example, *"I use Gnucash. Thus, I participate on documentation and website bugs for Gnucash, as it is the best use of my abilities. And I hope by doing this, some developer will consider fixing my bugs"*, and, *"I wasn't a big contributor at first but made a patch from time to time"*.

VCs understand they have to spend a significant amount of effort to make a difference and this is illustrated by, for example, *"I understand the fact that people are more likely to consider fixing the bugs that bug me if I make an effort to put something back into the community"*, and, *"My motivation in reporting issues is usually understanding that if I don't report the issue it will likely remain unresolved."*

In summary, opinions of respondents support our choice of the number and types of tasks and the effort to resolve these tasks as sensible measures of a participants willingness (and capacity). In particular, the amount of effort spent on tasks appears to be a good indicator of the willingness and it is a good discriminator between VCs and non-VCs. For example, one Gnome VC started his response with: *"I must admit I have been getting quite a lot of mails from researchers on various open source issues but nobody ever took the time to write individual mails to contributors and I really appreciate your effort"*. Most likely, whether or not these busy developers devote attention to a new participant, depends on the effort they see reflected in the activities of the newcomer.

VCs appear to have a positive engagement with their micro-environment, supporting both the measures we constructed and the impact of these measures we observed. In particular, VCs learned from their productive peers, for example, *"I learned a lot from this leading open source project while working with other brilliant contributors"*, providing evidence for the measure of peer's productivity, i.e., $PeerPerf$.

They were able (or lucky) to get attention from the community, for example, *"With bugzilla, the contribution is indirect, but the feedback from the developers shows that they care, and appreciate the effort I made, and actively work to solve the bug in a way that I can see progress. Eventually the bug will be fixed and released, and then the rest of the world can see it"*, supporting the positive impact of peer's attention, i.e., the negative impact of $LckAtten$ metric.

VSs were proud they were working on popular products, for example, *"As a volunteer free software developer I found it much more enjoyable to hack on code which I myself use everyday"*, and, *"Gnome is something which you can show to your friends and family members."*, supporting the use of macro-climate measures, in particular, product popularity ($nUsr$). However, the model shows $nUsr$ has a negative impact on newcomers' chance of becoming VCs. Given the fact that high $nUsr$ may overwhelm the mentors, newcomers may not be able to get enough attention and, therefore, feel unappreciated and leave the project, even though they join because of project's popularity. For example, as a Gnome VC explained: *"I try to comment on almost every bug filed explaining what I think of it and if I think it will be fixed soon or not. I also explain to people where in the code they might have a look to fix it to encourage them to help out."* However, *"we all have a life and time is limited. Often enough I find myself forgetting a bug report or even forgetting a patch and can kind of understand if people are pissed off by that."* Notice, $nUsr$ appears to be an important factor that keeps VCs to continue contributing, suggesting a positive impact that $nUsr$ may have on retaining VCs, a hypothesis that awaits a further study.

6 LIMITATIONS

We discuss some of the limitations related to the construction of measures and fitting models in Section 6.1, and specific issues encountered while conducting the prediction of VCs in Section 6.2. The internal and external validity are presented in Section 6.3.

6.1 Limitations of Bugzilla for Gnome and Mozilla

We start from the consistency and accessibility of the issue tracking data in the Bugzilla of Gnome and Mozilla.

First, we obtained the entire range of issues from 1 to 645899 for Mozilla and 639379 for Gnome (in 2011). Some

---

[18] This statement was verified by an inside developer. We asked one of the VCs (a survey respondent) what he thinks about this complain: *"(we all have a life and time is limited,) constantly annoying people is probably not the worst idea"*.

issues were either not public or not obtainable: 121578 in Gnome and 25388 in Mozilla. We compared different snapshots extracted at different times, and verified to make sure that the later snapshots include the issues in the earlier ones. While there were no problems obtaining Mozilla data, Gnome Bugzilla prevents retrieval of large numbers of issues with complete email address of contributors. We, therefore, had to rely on public extracts of Gnome Bugzilla data (i.e., Gnome 2006 extract and Gnome 2008 extract) to map logins to individuals.

Second, we assume that a login (for Gnome) and an email (for Mozilla) is a unique representation of a single person. However, multiple people may share the same login or a full email, and a single person may use several email addresses. To deal with this issue, we used two approaches. First, we identified "generic" logins, e.g., "mozilla", "gnome", "bugzilla*", "*maint" because we focused on ordinary contributors. Second, we used the full name of the participant associated with each login/email to identify all logins that had two or more names associated with them (multi-name logins). We also identified full names that had more than one login/email associated with them (multi-login names). To verify that our conclusions are not affected by this assumption, we fit our models on three datasets: the original data, the data without generic logins, and the data without generic and multi-name logins and with all logins having multi-login names replaced by the corresponding name. The results were similar and we reported results using the original data.

Third, the data itself might not reflect what actually happened. e.g., Gnome issue 572011 doesn't have an information page or an xml file, but it has an activity history page. Some MRs have some states missing. For example, for Mozilla issue 235354 the resolution type on the information page was "Status: RESOLVED NOTABUG", but on the history activity page the last resolution was INVALID. To address this limitation we tested how sensitive our analysis results are to these data consistency issues. 995 MRs in Gnome and 601 MRs in Mozilla had intermediates states missing, but excluding them didn't have any noticeable impact on the results.

We continue with the sensibleness of the measures we construct: do they reflect the intended concepts and can we measure each concept separately from other concepts? Individuals and their environment are notoriously difficult to measure because of the variability among individuals and the ambiguity of concepts such as willingness and opportunity. The issue tracking systems, however, provide a practical opportunity to observe the activities individuals engage in and infer the effort they spend to conduct them. From these basic measures it is possible to estimate their capacity and willingness.

Second, it is almost impossible to measure one separate dimension. People's activities we observe are always the combined effect of multiple dimensions. For example, the number of issues a participant is involved in shows not only her experience, but also her willingness to contribute, and, possibly, how buggy that product release is at that time. Fortunately, as Ericsson et al [6] claimed, ability and practice are not separable, i.e., the talent is not needed to explain performance if the amount of deliberate practice is taken into account and the only way to increase the amount of deliberate practice is through willingness. Therefore we encompass capacity and willingness into a single dimension — we refer to it as willingness — after all, willingness determines how much an individual would get involved in a volunteer activity.

At last, we get to the limitation that relates to the subjects we investigated. We investigated the participants' ability, willingness and environment when they joined the project. However, we noticed that some participants appear to join as developers instead of newcomers, i.e., commit a change to the VCS repository before showing up in Bugzilla. In particular, four people from the VC sample ($4/40 = 10\%$) whose issues we read, appear to have been in the development team from the very beginning. For example, the first participation of mozilla-22 is a comment: *"Checked in code for spec-compliant implementation of webclient on Solaris ..."*. This raises a question of how exactly those individuals became VCs, because the traces of their initial contributions are not recorded in Bugzilla. The survey responses provide some hints about this question. For example, some developer may have been with the project when the project was not open source — *"I worked at Netscape before the Mozilla project was started, and was one of the people who advocated for Netscape to release its source code for the browser"*, and *"For the first two or three years of my contributing to the project I was still a Netscape employee."* Because these VCs account for only 10% of all new VCs in our sample, and, to some extent, they do share many behaviors that characterize future VCs (e.g., commenting on more issues, showing more effort on reporting issues), we do not expect them to have a significant impact on the overall results. Meanwhile, an interesting question is what roles do those individuals play in cultivating the project's climate, and how they transfer their expertise to the newcomers.

6.2 Participant's Identity in Multiple Extracts

In theory, all the individuals identified in the Mozilla extract of 2011 should also be in the extract of 2012, because the later extract extends it. However, there were 459 emails in 2011 extract missing from 2012 extract. This was partly caused by the changes of email attribute associated with individual's key in the relational database used by the Bugzilla. Our data is based on the issue summary and history reports generated from Bugzilla, thus the email of the same participant may be different for reports retrieved at different times.

First, we observed occasions in which the same activities in the two extracts are associated with different emails. For example, Issue 490556 reporter is associated with *brent@aerobrent.com* in 2012 extract and with *aerobrent@gmail.com* in 2011 extract.

To address these potential inconsistencies we first obtained emails present in 2011 extract but not present in 2012 extract; Second, for each missing email, we located the associated activity in 2011 extract; Third, we located the same activity in 2012 extract based on the issue number, the type of activity, the order of the comment, and the issue attributes that were changed. We used actor's email in the corresponding activity to identify the change of email and associated both emails with the same individual. 439 out of 459 emails were linked this way.

The email change issue may present a problem for a single extract if, for example, a person changed email while we were retrieving the issues (the single extract took approximately two weeks to complete), so that we get one email for the issues retrieved before the change and another email for the remaining issues. It poses even more problems when comparing or merging two extracts. For example, in the 2011 extract, *Jason Duell* has *jduell@alumni.princeton.edu* associated with his activities in one issue report, but has *jduell.mcbugs@gmail.com* associated with the remaining activities in other reports. To confirm that it was the database change, we verified in the 2012 extract that *jduell@alumni.princeton.edu* was replaced by *jduell.mcbugs@gmail.com*. There were three individuals whose email changed while we were obtaining the 2011 extract.

We also discovered additional problems with the data. For example, among 459 emails missing from the 2012 extract, we found 17 to be mentioned in Issue 452498. As it turned out, there was an error while generating or retrieving the issue report. In the 2011 extract, all the activities related to Issue 452498 had only login instead of the full email. For example, instead of *shaver@mozilla.org* there was just *shaver*. This suggests that the retrieval script has not successfully signed in into Bugzilla before retrieving the issue, because Bugzilla reports for unauthenticated users are generated without including the full email.

6.3 Internal and External Validity

From the internal validity perspective we checked the assumptions for the logistic regression and log-transformed the predictors. While only $19\%(23\%)$ of the deviance is explained, this is, in fact, a good fit given that only $0.9\%(3.6\%)$ of the participants become VCs in Mozilla (Gnome). From the prediction perspective, among the 25,406 Mozilla participants whose predicted VC probability is above 99-th percentile, 24% are VCs and 47% of VCs are predicted. For comparison, for a random predictor the precision would be 0.516% or 72 times lower, and the recall would be 1% or 47 times lower.

From the external validity perspective, the way Gnome and Mozilla are operating is not unusual for open source projects. The models for both projects are quite similar. However, both are large projects and both represent user interface domain. We, therefore, may not generalize to other domains (e.g., server), and smaller projects.

It's important to stress that our findings show association between the response and predictors, but that association may not be causal. In particular, there may be some aspects of individual character or of the environment that we did not measure, but that cause both the response and the predictors to behave in the observed pattern.

7 CONCLUSIONS

In this study, we address the following question: what impacts the chances that a joiner to a software project will become a VC? Are these probabilities affected by people's character, project's climate, or, the interaction between joiners and their environments? We measured the behavior of individual participants in Gnome and Mozilla using issue workflow, and modeled how the differences in their behavior affected the probability of the participant becoming a VC.

We found that the main differences among participants were in their capacity, willingness and opportunity to contribute at the time of joining. A participant's capacity and willingness are measured through the activities she takes on, i.e., the number and type of tasks she starts with, and the effort she puts into her contributions, e.g., the fraction of reported issues that were ultimately fixed. The opportunity is measured via environment's macro-climate that was shared among all participants and micro-climate that was unique for a person. In particular, product's popularity, project's task density, and project's sociality constitute measures of macro-climate, while the initial size of peer group, peers' performance, their social clustering, and their attention to the new participant constitute micro-climate.

We found the probability of a newcomer becoming a VC to be associated with person's willingness and environment. Most importantly, her pro-community attitude represented by her first contribution being a comment on an existing issue instead of a bug report or a report through Bugzilla interface instead of a crash-reporting tool double her odds of becoming a VC. Having at least one issue reported during the first month to be fixed has the same effect. Her micro-climate represented by low attention of a too-rapid response to issue reports, and her macro-climate represented by the increased project popularity reduce her odds. Project's relative sociality and individual's peer network size had opposite effects in the two projects. This may reflect some inherent differences between practices of Gnome and Mozilla that need further study.

These findings may help individual participants to understand what their own roles are and find the best ways to contribute. It is also likely to help OSS communities to adopt better strategies to attract and retain newcomers. Specifically, the probability of staying longer is associated with how much value a new participant provides to the project by commenting, putting more effort into issue reports, and by the amount of attention the project provides to the newcomer. Ironically, it is during times when projects are the most popular, with a stream of joiners overwhelming the mentors, the community needs to put extra effort to retain newcomers. However, the opinions from the survey respondents indicate that there is a trade-off between attracting newcomers and the workload of existing contributors. How to achieve this balance is worth further study as noted by a disappointed contributor *"If you know a way of micro-contributing which don't leave feeling that nobody cares, let me know."*

8 ACKNOWLEDGMENT

Partial support by National Basic Research Program of China Grant 2009CB320703 and the National Natural Science Foundation of China Grants 91118004, 61121063 and 61073016.

REFERENCES

[1] M. Zhou and A. Mockus, "Developer fluency: Achieving true mastery in software projects," in *ACM SIGSOFT / FSE*, Santa Fe, New Mexico, November 7–11 2010, pp. 137–146. [Online]. Available: papers/fluency.pdf

[2] K. R. Lakhani and E. von Hippel, "How open source software works: free user-to-user assistance," *Research Policy*, vol. 32, no. 6, pp. 923 – 943, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0048733302000951

[3] M. Blumberg and C. D. Pringle, "The missing opportunity in organizational research: Some implications for a theory of work performance," *The Academy of Management Review*, vol. 7, no. 4, pp. pp. 560–569, 1982. [Online]. Available: http://www.jstor.org/stable/257222

[4] M. Zhou and A. Mockus, "Does the initial environment impact the future of developers?" in *ICSE 2011*, Honolulu, Havaii, May 21–28 2011, pp. 271–280.

[5] U. Neisser, G. Boodoo, J. Bouchard, Thomas J., A. W. Boykin, N. Brody, S. J. Ceci, D. F. Halpern, J. C. Loehlin, and R. Perloff, "Intelligence: Knowns and unknowns," *American Psychologist*, vol. 51, no. 2, pp. 77–101, Feb. 1996.

[6] K. A. Ericsson, R. T. Krampe, and C. Tesch-Rmer, "The role of deliberate practice in the acquisition of expert performance," *Psychological Review*, vol. 100, no. 3, pp. 363–406, Jul. 1993.

[7] H. M. Parsons, "What happened at hawthorne?" *Science*, vol. 8, no. 4, pp. 922–932, 8 March 1974.

[8] B. Curtis, "Fifteen years of psychology in software engineering: Individual differences & cognitive science," in *ICSE'84*, 1984, pp. 97–106.

[9] J. D. Couger and R. A. Zawacki, *Motivating and Managing Computer Personnel*. New York, NY, USA: John Wiley & Sons, Inc., 1980.

[10] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, and Y. Ye, "Evolution patterns of open-source software systems and communities," in *IWPSE '02: Proceedings of the International Workshop on Principles of Software Evolution*, Orlando, FL, May 19–20 2002, pp. 76–85.

[11] A. Mockus, "Organizational volatility and its effects on software defects," in *ACM SIGSOFT / FSE*, Santa Fe, New Mexico, November 7–11 2010, pp. 117–126. [Online]. Available: papers/orgQuality.pdf

[12] A. Ko, R. DeLine, and G. Venolia, "Information needs in collocated software development teams," in *ICSE 2007*. ACM Press, May 20–26 2007, pp. 344–353.

[13] M. Cataldo, P. Wagstrom, J. Herbsleb, and K. Carley, "Identification of coordination requirements: Implications for the design of collaboration and awareness tools." in *Conference on Computer Supported Cooperative Work CSCW'06*, Banff, Alberta, Canada, 2006.

[14] B. Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. P. de Vrie, "Moving into a new software project landscape," in *ICSE 2010*, Cape Town, South Africa, May 1-8 2010, pp. 275–284.

[15] N. Ducheneaut, "Socialization in an open source software community: A socio-technical analysis," *Journal of Computer Supported Collaborative Work*, vol. 32, pp. 323–368, 2005.

[16] A. Mockus, R. T. Fielding, and J. Herbsleb, "Two case studies of open source software development: Apache and Mozilla," *ACM Transactions on Software Engineering and Methodology*, vol. 11, no. 3, pp. 1–38, July 2002. [Online]. Available: papers/mozilla.pdf

[17] D. M. German, "The gnome project: a case study of open source, global software development," *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 201–215, 2003.

[18] P. A. Wagstrom, "Vertical interaction in open software engineering communities," *Phd thesis, Carnegie Mellon University*, vol. CMU-ISR-09-103, March 2009.

[19] J. Xie, A. Mockus, and M. Zhou, "Visualizing evolution of software issue-tracking practices," in *ESEM 2011*, Banff, Canada, Sep 22–23 2011.

[20] P. Robillard, "The role of knowledge in software development," *Communications of the ACM*, vol. 42, no. 1, pp. 87–92, 1999.

[21] J. Lerner and J. Tirole, "Some simple economics of open source," *The Journal of Industrial Economics*, vol. 50, no. 2, pp. 197–234, 2002. [Online]. Available: http://www.jstor.org/stable/3569837

[22] L. L. Cummings and D. P. Schwab, "Performance in organizations: Determinants and appraisal." *Administrative Science Quarterly*, vol. 18, no. 3, pp. pp. 412–414, 1973. [Online]. Available: http://www.jstor.org/stable/2391680

[23] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *The Academy of Management Review*, vol. 20, no. 3, pp. pp. 709–734, 1995. [Online]. Available: http://www.jstor.org/stable/258792

[24] A. Mockus, "Succession: Measuring transfer of code and developer productivity," in *2009 International Conference on Software Engineering*. Vancouver, CA: ACM Press, May 12–22 2009. [Online]. Available: papers/succession.pdf