We examined the mailing lists from the community and for each developer counted the number of messages posted, new discussion threads started, projects mailing lists they were active on, and the number of projects they posted messages to for which they had never contributed code. Each of these values was normalized by the number of years the developer had been in the community, as measured by the duration from their first observable contribution in any project to their last observable contribution (or the end of the data set if still active). In addition we examined the messages posted by the developers looking for pro-social behaviors such as providing references to another developer's email address and post links to websites, indicating that the developer was attempting to redirect and guide other individuals. We then performed an ANOVA to compare among the classes of developers: volunteer, product focused, and community focused. The results as shown in Table 3 indicate that there is a significant difference in participation patterns between the three classes of developers. In particular, commercial developers were found to be much more active on mailing lists. However, when tests were performed analyzing just the difference between product and community focused commercial developers; a difference was found only in the number of messages posted to project mailing lists.

Insert Table 3 about here

Further content analysis of all email messages sent to public mailing lists identified elements that support information seeking behavior in the community – posting email addresses of contacts and providing pointers to web pages. The results were then aggregated by whether the author of the message was employed by a community or product focused firm. This analysis found that community focused developers posted references to email addresses 86% more frequently than product focused developers, and web addresses 161% more often. Both of these behaviors are pro-social community building behaviors that help new community members become acquainted with the project and participants.

Mid-level technical interactions on the Bugzilla bug tracking system may have similar affects in building community as posting to message to a mailing list. In particular, users are encouraged to post any bugs encountered to Bugzilla. These bugs are periodically triaged by a group of community members who then assign the bugs to project developers. At the simplest level, each bug is given a small message form that allows developers to post messages that are sent back to the original submitter and any other individual with interest in the bug. Often times, developers post messages indicating that a bug has been verified as present, asking for more information, or provide a workaround for the user experiencing the bug. Individuals also may submit patches to bugs that address the bug, a behavior that could be seen by a volunteer as taking a significant interest in their issue. When working with Bugzilla, developers can mark bugs as "fixed", indicating that the patch has been submitted and accepted. Finally, we can count the number of distinct projects a developer was active on within the Bugzilla system to get an idea of overall activity and also cross-reference this against activity in the source code repository to see where developers contribute to bug management but do not write code. We take this to be an indication that a developer is taking concrete action showing a broader concern for the project, beyond the local areas that are the focus of the developer's interest.

We collected each of the previously described metrics for every developer and used the same method as we did for the mailing lists to normalize for the length of time the developer was active in the community. The metrics were aggregated by class of developer, and summarized in
Table 4. Surprisingly, while commercial developers had a greater frequency of activity, as measured by the number of comments, patches, and bugs fixed, they had the same relative amount of breadth in the system as volunteer developers. Contrary to our initial belief, when accounting for tenure in the project, product focused developers were active on project bug trackers for significantly more projects and projects for which they had written no code than community focused developers (as measured by the Extra Projects variable). However, as a whole, the ANOVA for these values were not significant.

Insert

Table 4 about here

Certain actions by commercial developers in the CVS code repository may also be construed as pro-social community oriented behavior. Working on a variety of projects within the community shows that the developer has a greater interest in the overall health and well-being of the community. Analysis of the logs indicates that developers employed by community focused firms are active on significantly more projects, as shown in Table 5. The increased participation across a wider number of projects confirms the responses by many of the volunteer interviewees who believed the product focused firms worked only in narrow niches within the community and that community focused firms spread their effort across multiple projects.

Insert Table 5 about here

This analysis shows that there are sometimes dramatic differences in the patterns of participation between volunteer, product focused, and community focused developers. While, in general, all commercial developers are more active in the community than volunteer developers, community focused developers are much more active and visible on community mailing lists and within the project source code. However, product focused developers tend to be more active in the moderately technical realm of bug tracking, indicating that such work may be critical to the success of their business model.

Quantifying the Impact of Commercial Developers on Volunteer Participation

The community makes it very easy for developers to start a new project, leading to a variety of projects that contain only small amounts of code, or represent the efforts of only a single developer working on a very specific tool. To select projects that had a substantial community around them and enough data to achieve statistical significance, we filtered the data selecting only projects with more than 20 developers, more than 100 bugs filed in the Bugzilla bug tracking system, at least one community hosted mailing list

associated with the project, and more than a year of overlap between the source code history, mailing list archives, and bug tracker data. These requirements yielded 14 projects from the community. To avoid problems related to temporary changes in behavior, for example, the "freezing" of the project code right before a release when only critical fixes are allowed into the project source code repository, the data was broken up into 8 week periods. At each period the number and identity of volunteer and commercial developers committing code during the period and the number of commits to the project during the period were recorded. Summary statistics and correlations can be seen in Table 6 and Table 7 below. The distribution of the number of commercial and volunteer users is highly skewed toward the lower end of the range and can be approximated with a log-normal distribution. To a lesser degree the distributions of the number of community focused developers, product focused developers, and commits are also skewed. To accommodate for this in our models, the logarithm (base 2) of these variables is used. Of note regarding the correlations is the high correlation between the number of volunteer developers at time t and time $t-1$. This is a sign of a broader problem of autocorrelation in the number of volunteer developers across many time periods, which can be seen in Figure 1. This high level of autocorrelation can be compensated for by examining the difference in the number of volunteer developers between different time periods. The autocorrelation of this new variable is seen in Figure 2. The autocorrelations are now significantly lower, however there remains a moderately high autocorrelation with a time lag of one. This indicates that periods with high volunteer influx will often be followed by volunteer outflow in subsequent periods. We consider this correlation of -0.25 as sufficiently low and only have negligible impact on the regression models in Equation 1 and 2.

Insert Table 6 about here

Insert Table 7 about here

Insert Figure 1 about here

Insert Figure 2 about here

We begin with a regression model that predicts the change in the logarithm of the number of volunteer developers contributing source code to project at time t as a function of the log transformed number of volunteer developers, commercial developers, and commits at time $t-1$. To accommodate for the varying level of inherent attractiveness for different projects in the ecosystem, each project in the regression was represented by an indicator variable, α_i, to fit the so-called fixed-effects model. The regression model is shown in Equation 1. Within this model, the response variable is the difference in the log of the number of volunteer developers, which is functionally the same as the percentage change in the number of volunteer developers, for project i from time period $t-1$ to t, $\text{diff}(\log(VolDevs_{i,t}))$ and the predictor variables are the intercept for the project, α_i, the log of the number of volunteer developers for project i in the previous time period, $\log(VolDevs_{i,t-1})$, the log of the number of commercial developers for project i in the previous time period, $\log(ComDevs_{i,t-1})$, the number of commits to the project version control system for project i in the previous time period, $\log(Commits_{i,t-1})$, and an identifier for the current time period, $\log(t)$.

Equation 1:

$$\text{diff}(\log(\boldsymbol{VolDevs_{i,t}})) = \boldsymbol{\alpha_i} + \boldsymbol{\beta_0}\log(\boldsymbol{VolDevs_{i,t-1}}) + \boldsymbol{\beta_1}\log(\boldsymbol{ComDevs_{i,t-1}}) + +\boldsymbol{\beta_2}\log(\boldsymbol{Commits_{i,t-1}}) + \boldsymbol{\beta_3}\log(\boldsymbol{t}) + \boldsymbol{\epsilon_{i,t}}$$

The results of the model, reported in Table 8 indicate that an increase in the number commercial software developers working on a project has no effect on attracting additional volunteer developers to the project. However, general activity in a project, as measured by the number of commits, is related to an increase in the number of volunteer developers in the next time period. This effect is tempered by the fact that project growth typically slows down relative to the size of the project as project ages, and therefore, they attract

fewer new volunteers relative to their existing size. Coefficients for project indicator variables, not shown, ranged from 0.09 to 0.89 and were significant at the $p < 0.001$ level for 11 of the 13 projects. Due to the lack of significance of $\log(ComDevs_{i,t-1})$ we conclude there is not support for hypothesis 1 or hypothesis 2.

Insert Table 8 about here

As we have shown there is a marked difference between the methods and magnitudes of participation of the two types of commercial developers: product focused and community focused. In Equation 2 we expand on the model through the use of firm type as a moderating variable to differentiate between participation of developers for community focused firms, $ComDevs_{CF,t-1}$, and product focused firms $ComDevs_{PF,t-1}$. The same data are used with the new multi-level longitudinal model, with the regression coefficients presented in Table 9.

Equation 2:

$$\begin{aligned}\text{diff}(\log(\boldsymbol{VolDevs_{i,t}})) &= \boldsymbol{\alpha_i} \\ &+ \boldsymbol{\beta_0}\log(\boldsymbol{VolDevs_{i,t-1}}) + \boldsymbol{\beta_1}\log(\boldsymbol{ComDevs_{CF_{i,t-1}}}) + \boldsymbol{\beta_2}\log(\boldsymbol{ComDevs_{PF_{i,t-1}}}) \\ &+ \boldsymbol{\beta_3}\log(\boldsymbol{Comits_{i,t-1}}) + \boldsymbol{\beta_4}\log(\boldsymbol{t_i}) + \boldsymbol{\epsilon_{i,t}}\end{aligned}$$

Insert Table 9 about here

In contrast to the original model where there was no significant relationship between firm participation and the change in the number of volunteers, when the firms are classified by business model, there is a significant difference. Participation by developers from community-focused firms has a significant and positive relationship to the change in the number of volunteer users, while participation by developers for product focused firms has no statistically significant impact. The other coefficients in the model remain similar to those shown for the more basic model shown in Table 8 and the explanatory power of the

model has increased slightly. This difference between developers at community and product focused firms lends support for hypothesis 3.

In order to evaluate Hypothesis 4, we needed to subdivide projects into their constituent modules, and use modules, rather than projects, as our unit of analysis. While language specific methods exist to specify explicit module structures and infer implicit structure through static source code analysis, the code in the GNOME project is written in a variety of different languages and programming language specific methods are inconsistent and impractical. As an alternative, we used social network analytic clustering methods on the network of source code to approximate modules within the project. While a variety of unsupervised clustering methods exist that heuristically determine an optimal number of clusters (e.g. Newman, 2004) these methods often produce more clusters than practical, leaving many clusters with only a single active developer. The CONCOR algorithm was used as it requires no additional information beyond link information when generating the groupings and groups elements based on structural similarity. Functionally, the computation views the network of files as a matrix and then attempts to rearrange the rows and columns of the matrix so entities that are structurally equivalent, meaning they link to the same set of other files, are grouped together (Boorman & White, 1976).

To obtain the network for CONCOR clustering, we utilized logical dependency links between files: nodes in the network are files, and edges are added between nodes if they were committed back to the central repository in a single commit. This approach is commonly used in software engineering research and is based on the observation that files are committed together by the same person frequently have dependencies on each other and is suitable for situations where language specific methods fail (Gall, Hajek, & Jazayeri, 1998). In this way, we obtain a network where highly related files are densely clustered together. The CONCOR algorithm was run on this network for each project and configured to generate 8 clusters, each approximating a module within the project. As CONCOR is a binary slicing

algorithm we needed to pre-specify the number of groups for each project – eight was chosen as a compromise for very large projects, for which four groups would result in many disparate modules being lumped together, and smaller projects, for which there were not sixteen different components. The same summary statistics shown in Table 6 were generated for each module in each eight week time period, yielding a total of 6360 observations.

The analysis at the project level was then replicated with the new data based on the clusters within each project. For this analysis, a new subscript j is added to the model indicating the cluster within project ii. Intercepts are calculated for each of the clusters in the data, $\alpha_{i,j}$, and time is the number of 8-week periods since the start of the project. The complete model is shown in Equation 3 and the results appear in Table 10.

Equation 3:

$$\begin{aligned}\text{diff}\left(\log\left(\boldsymbol{VolDevs}_{i,j,t}\right)\right) &= \boldsymbol{\alpha}_{i,j} \\ &+ \boldsymbol{\beta_0}\log\left(\boldsymbol{VolDevs}_{i,j,t-1}\right) \\ &+ \boldsymbol{\beta_1}\log\left(\boldsymbol{ComDevs}_{i,j,t-1}\right) + \boldsymbol{\beta_2}\log\left(\boldsymbol{Commits}_{i,j,t-1}\right) + \boldsymbol{\beta_3}\log(\boldsymbol{t}_i) + \boldsymbol{\epsilon}_{i,j,t}\end{aligned}$$

Insert Table 10 about here

The effects of the model largely mirror the results found when analyzing at the project level (see Table 8). The lack of significance for the coefficient of $\log\left(ComDevs_{i,j,t-1}\right)$ shows a lack of support for Hypothesis 4. As before, we examine the difference in the effect of commercial developers when the developers are segregated by firm. This differentiation results in a new model shown in Equation 4 and the results are presented in Table 11.

Equation 4:

$$\begin{aligned}\text{diff}\left(\log\left(VolDevs_{i,j,t}\right)\right) &= \alpha_{i,j} \\ &+ \beta_0 \log\left(VolDevs_{i,j,t-1}\right) \\ &+ \beta_1 \log\left(ComDevs_{CF_{i,j,t-1}}\right) \\ &+ \beta_2 \log\left(ComDevs_{PF_{i,j,t-1}}\right) + \beta_3 \log\left(Commits_{i,j,t-1}\right) + \beta_4 \log(t_i) + \epsilon_{i,j,t}\end{aligned}$$

Insert Table 11 about here

Under this new model we see that at the module level community focused developers once again are related to an increase in volunteer participation and that product focused developers have a significant negative relation to volunteer participation. This paints a mixed picture regarding the increase in cognitive complexity at the module level. For product focused developers there is support for hypothesis 4. However, community focused developers retain their positive relationship with the change in the number of volunteer developers, indicating that whatever cognitive complexity they introduce may be overcome by their general power to attract new developers to the community. Therefore, we reject hypothesis 4 for community focused developers.

DISCUSSION

As Open Source continues to grow, more commercial developers will need to find ways to work with pre-existing communities of volunteers and other commercial developers. While there are numerous ways that commercial developers can affect the level of volunteer and independent participation in these projects, this work establishes that in terms of the number of volunteer developers, there is little overall negative impact from commercial developers participating in existing Open Source communities. At the aggregate we observe that commercial developers have no statistically significant impact on the change in volunteer developers, which does not allow us to confirm or reject hypotheses 1 and 2. For Open Source community managers and project managers, this should be good news as it indicates that their population

of volunteers will most likely remain steady in the presence of commercial interests utilizing and modifying the project code for commercial purposes.

Another major contribution is the identification and validation of two distinct classes of business models for firms participating in Open Source communities and the different impact they have on volunteer participation. These models closely aligned themselves either with broad community success or with a single project in the community. As expected, community focused firms were found to be highly visible on project mailing lists and wrote code for more projects than developers from product focused firms. Yet, contrary to our assumptions, product focused firms were more active in the moderately technical domain of bug reporting and management. The different business models were found to have very different impacts on volunteer developers – with community focused developers attracting volunteers while product focused developers had no statistically significant effect, supporting hypothesis 3. Given the predominant view of the interviewees that community focused firms were more aligned with the values and norms of the community, this supports the notion that the communities are sensitive to the values and norms of commercial participations. This result should urge caution on firms wishing to participate in Open Source projects, and suggests that behaving in a way that supports the community may actually strengthen and enlarge it.

Finally, we analyzed whether or not commercial developers had a negative effect on volunteer participation at the module level, possibly due to some of the properties of commercial development such as developer collocation, increased ability to work with others, or increased and technical skill. Much to our surprise it was found that this was the case for developers from product focused firms, but not community focused firms. This indicates that there is the possibility the commercial firms working on a module within a large Open Source project may drive volunteer developers away, as is the case with product focused firms. However, this doesn't need to be the case, the differences in behavior for

community focused developers along with their general attractive power seem to negate these negative effects and once again show are correlated with an increase in the number of volunteer developers.

Our research suggests both caution and some reassurance for firms considering a product-focused relationship to an Open Source community. Our qualitative results show that volunteer developers frequently made negative comments about product focused firms, which is quite worrisome. On the other hand, increased participation of developers from product-oriented firms did not drive volunteers away. It may be that the increased visibility that commercial participation lends a project offsets any negative effects from its perceived failure to uphold community norms.

No matter what the reasons for the increased success of community focused firms in attracting and retaining volunteer developers, it appears that more and more firms will adopt Open Source projects, practices, and contribute to communities in the near future. We have seen that in this case, the dual worlds of volunteer and commercial can co-exist in an Open Source project with little danger of the commercial firm dramatically damaging the incumbent volunteers. Going forward, understanding the methods by which these firms attract and retain volunteer developers is an open research question that will yield great benefits for firms seeking to utilize this revolutionary software development model.

BIBLIOGRAPHY

Ashforth, B. E., & Mael, F. 1989. Social Identity Theory and the Organization. ***The Academy of Management Review***, *14*(1): 20-39.

Baldwin, C. Y., & Clark, K. B. 2000. ***Design Rules, Vol. 1: The Power of Modularity*** (First Edition.), Cambridge, MA: The MIT Press.

Bartholomew, D. 2008. A look at the Kindle. ***Linux Journal***, *2008*(176): 3.

Barton, J. 2003. From Server Room to Living Room. ***Queue***, *1*(5): 20-32.

Bitzer, J., Schrettl, W., & Schroder, P. J. 2007. Intrinsic motivation in open source software development. ***Journal of Comparative Economics***, *35*(1): 160-169.

Bonaccorsi, A., Giannangeli, S., & Rossi, C. 2006. Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry. ***Management Science***, *52*(7): 1085-1098.

Boorman, S. A., & White, H. C. 1976. Social Structure from Multiple Networks. II. Role Structures. ***American Journal of Sociology***, *81*(6): 1384-1446.

Brooks, F. P. 1995. ***The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition*** (2nd ed.), Boston, MA, USA: Addison-Wesley Professional.

Campagnolo, B., Tacla, C. A., Paraiso, E. C., Sato, G. Y., & Ramos, M. P. 2009. An architecture for supporting small collocated teams in cooperative software development. In ***International Conference on Computer Supported Cooperative Work in Design***: 264-269, Los Alamitos, CA, USA: IEEE Computer Society.

Fielding, R. T. 2005. Software architecture in an open source world. In ***27th International Conference on Software Engineering***: 43, Presented at the 2005 International Conference on Software Engineering (ICSE 2005), St. Louis, MO, USA.

Fielding, R. T. 1999. Shared leadership in the Apache project. ***Communications of the ACM***, *42*(4): 42-

43.

Fogel, K. 2005. ***Producing Open Source Software***, Sebastapol, CA: O'Reilly & Associates.

Gall, H., Hajek, K., & Jazayeri, M. 1998. Detection of Logical Coupling Based on Product Release History. In ***14th IEEE International Conference on Software Maintenance***, Presented at the International Conference on Software Maintenance, Bethesda, MD: IEEE Press.

German, D. 2005. Software Engineering Practices in the GNOME Project. In J. Feller, B. Fitzgerald, S. A. Hissam, K. R. Lakhani, & M. Cusumano (Eds.), ***Perspectives on Free and Open Source Software***: 211-226, Cambridge, MA: MIT Press.

German, D. 2004. The GNOME project: a case study of open source, global software development. ***Software Process: Improvement and Practice***, *8*(4): 201-215.

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. 2002. ***Free/Libre and Open Source Software: Survey and Study***, International Institute of Infonomics University of Maastricht, The Netherlands.

Goldman, R., & Gabriel, R. P. 2005. ***Innovation Happens Elsewhere: Open Source as Business Strategy***, San Francisco, CA: Morgan Kaufmann.

Hars, A., & Ou, S. 2002. Working for Free? Motivations for Participating in Open-Source Projects. ***International Journal of Electronic Commerce***, *6*(3): 25-39.

Hertel, G., Niedner, S., & Hermann, S. 2003. Motivation of Software Developers in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel. ***Research Policy***, *32*(7): 1159-1177.

de Icaza, M. 2002, April 9. The Story of the GNOME Project, http://primates.ximian.com/~miguel/gnome-history.html, January 10, 2010.

Jackson, S. E., Brett, J. F., Sessa, V. I., Cooper, D. M., Julin, J. A., & Peyronnin, K. 1991. Some differences make a difference: Individual dissimilarity and group heterogeneity as correlates of recruitment, promotions, and turnover. ***Journal of Applied Psychology***, *76*(5): 675-689.

Koch, S., & Schneider, G. 2002. Effort, co-operation and co-ordination in an open source software

project: GNOME. ***Information Systems Journal***, *12*(1): 27-42.

Krishnamurthy, S. 2005. An Analysis of Open Source Business Models. In J. Feller, B. Fitzgerald, S. Hissam, & K. R. Lakhani (Eds.), ***Perspectives on Free and Open Source Software***, Cambridge, MA: MIT Press.

Lakhani, K., & Wolf, R. 2005. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In Joseph Feller, Brian Fitzgerald, Scott Hissam, & Karim R. Lakhani (Eds.), ***Perspectives on Free and Open Source Software***, Cambridge, MA: MIT Press.

Lawton, G. 2008. US Cell Phone Industry Faces an Open Future. ***Computer***, *41*(2): 15-18.

MacCormack, A., Rusnak, J., & Baldwin, C. Y. 2006. Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. ***Management Science***, *52*(7): 1015-1030.

Maxwell, E. 2006. Open Standards, Open Source, and Open Innovation: Harnessing the Benefits of Openness. ***Innovations: Technology, Governance, Globalization***, *1*(3): 119-176.

Moody, G. 2001. ***Rebel Code: Linux and the Open Source Revolution***, New York, NY: Basic Books.

Neus, A., & Scherf, P. 2005. Opening minds: Cultural change with the introduction of open-source collaboration methods. ***IBM Systems Journal***, *44*(2): 215-225.

Newman, M. 2004. Detecting community structure in networks. ***The European Physical Journal B***, *38*(2): 321-330.

Oreg, S., & Nov, O. 2008. Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. ***Computers in Human Behavior***, *24*(5): 2055-2073.

Parnas, D. 1972. On the Criteria to be Used in Decomposing Systems Into Modules. ***Communications of the ACM***, *15*(12): 1053-1058.

Pelled, L. H., Eisenhardt, K. M., & Xin, K. R. 1999. Exploring the Black Box: An Analysis of Work Group Diversity, Conflict, and Performance. ***Administrative Science Quarterly***, *44*(1): 1-28.

Polletta, F., & Jasper, J. M. 2001. Collective Identity and Social Movements. ***Annual Review of Sociology***, *27*(1): 283-305.

Raymond, E. S. 1999. ***The Cathedral and the Bazaar***, Sebastapol, CA: O'Reilly & Associates.

Roberts, J. A., Hann, I., & Slaughter, S. A. 2006. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. ***Management Science***, *52*(7): 984-999.

Rosen, L. 2004. ***Open Source Licensing: Software Freedom and Intellectual Property Law***, Upper Saddle River, NJ: Prentice Hall PTR.

Shah, S. K. 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. ***Management Science***, *52*(7): 1000-1014.

Star, S. L., & Griesemer, J. R. 1989. Institutional Ecology, `Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. ***Social Studies of Science***, *19*(3): 387-420.

Stewart, K. J., & Gosain, S. 2006. The Impact of Ideology on Effectiveness in Open Source Software Development Teams. ***MIS Quarterly***, *30*(2): 291-314.

The GNOME Foundation. 2010, January. GNOME: The Free Software Desktop Project, http://www.gnome.org/, January 6, 2010.

Tiemann, M. 1999. Future of Cygnus Solutions: An Entrepreneur's Account. In C. DiBona, S. Ockman, & M. Stone (Eds.), ***Open Sources: Voices from the Open Source Revolution***: 71-91, Sebastapol, CA: O'Reilly Media, Inc.

Williams, K., & O'Reilly, C. 1998. Demography and diversity in organizations: A review of 40 years of research. ***Research in Organizational Behavior***, *20*: 77-140.

Table 1: General Description of Interviewees

Total Interviewees	18
Commercial Developers	50%
Volunteer Developers	50%
Student Volunteer Developers	17%
Commit Access	78%
Self-Described as Developer	89%
Self-Described as Community Support	11%
Longest Participation	10 years
Shortest Participation	11 months
Median Participation	3 years

Table 2: Major firms participating in the community

Product Focused Firms		Community Focused Firms	
Firm A	A large IT firm that became involved through the purchase of Firm B and has greatly expanded to a wide variety of desktop applications and a distribution of Linux.	Firm F	A large Linux distributor that distributes GNOME and sells and supports Linux for the desktop and server.
Firm B	A medium firm that developed enterprise class desktop applications and system support for the GNOME. Purchased by Firm A.	Firm G	A large IT firm that uses GNOME as a compliment to its hardware offerings.
Firm C	A small firm that specializes in adapting small portions GNOME for embedded applications.	Firm H	A Linux distributor that historically shipped a desktop environment from a competing Open Source community and had small participation in the community.
Firm D	A small firm that produced enterprise class desktop applications for the GNOME community.	Firm I	A medium Linux distributor that historically supported both GNOME and another community with a similar goal.
Firm E	A small venture capital funded firm that developed software and sold integrated services for the GNOME community		

Table 3: Mean Activity per Year on Mailing Lists by Class of Developer (superscripts indicate statistically different groups of means in each row)

Variable	Volunteer Mean	Product Focused Mean	Community Focused Mean	P Value
Messages	39.04^{A}	87.10^{B}	135.80^{C}	<0.001
Threads Started	13.85^{A}	34.42^{B}	56.37^{B}	<0.001
Mailing Lists	0.37^{A}	0.68^{B}	0.53^{B}	<0.001
Extra Mailing Lists	0.20^{A}	0.23^{A}	0.20^{A}	0.400
Email References/Message	0.22^{A}	0.22^{A}	0.41^{B}	<0.001
URLs/Message	0.48^{A}	0.31^{B}	0.81^{C}	<0.001

Table 4: Mean Activity per Year in Bug Reporting Database by Class of Developer (superscripts indicate statistically different groups of means in each row)

Variable	Volunteer Mean	Product Focused Mean	Community Focused Mean	P Value
Comments	74.92^{A}	156.50^{B}	133.40^{B}	0.010
Patches	4.93^{A}	9.60^{A}	6.14^{A}	0.042
Bugs Fixed	1.44^{A}	3.80^{B}	7.30^{B}	<0.001
Projects	2.60^{A}	3.54^{B}	2.43^{A}	0.141
Extra Projects	1.56^{A}	2.09^{B}	1.11^{A}	0.556

Table 5: Mean Activity per Year in CVS Repository by Class of Developer (superscripts indicate statistically different groups of means in each row)

Variable	Volunteer Mean	Product Focused Mean	Community Focused Mean	P Value
CVS Projects	13.72^{A}	5.24^{B}	15.29^{A}	0.002

Table 6: Summary Statistics of Data Collected from 14 projects at 8 week intervals (601 total observations)

	Mean	Median	Skewness	Kurtosis	Std Dev	Max	Min
$VolDevs_{i,t}$	4.01	3	1.24	0.96	3.94	18	0
Number of volunteer developers contributing code to project i at time t							
$ComDevs_{i,t}$	3.57	2	2.19	4.70	4.87	26	0
Number of commercial developers contributing code to project i at time t							
$ComDevs_{CF,i,t}$	1.12	1	2.16	4.77	1.66	9	0
Number of commercial developers from community focused firms contributing code to project i at time t							
$ComDevs_{PF,i,t}$	2.65	1	2.84	8.22	4.49	26	0
Number of commercial developers from product focused firms contributing code to project i at time t							
$Commits_{i,t}$	114.97	46	2.98	11.54	175.64	1407	0
Number of commits made by all developers to project i at time t							
Observations per Project, N	42.92	49	-1.66	1.74	12.72	53	14

Table 7: Correlations of Data Collected at Project Level after $\log_2$ transformations.

	$VolDevs_t$	$VolDevs_{t-1}$	$ComDevs_{t-1}$	$ComDevs_{CF,t-1}$	$ComDevs_{PF,t-1}$
$VolDevs_{t-1}$	0.8263				
$ComDevs_{t-1}$	0.3755	0.3921			
$ComDevs_{CF,t-1}$	0.4346	0.4258	0.6272		
$ComDevs_{PF,t-1}$	0.2733	0.2773	0.9272	0.3555	
$Commits_{t-1}$	0.6655	0.7331	0.6400	0.4208	0.5504

Table 8: Hypothesis 1 and 2 – Regression Coefficients Predicting Change in Number of Volunteer Developers by Project (Equation 1)

Variable	Estimate	Std Err	P Value
$\log(VolDevs_{i,t-1})$	-0.4779	0.0356	<.001
$\log(ComDevs_{i,t-1})$	0.0411	0.0311	0.187
$\log(Commits_{i,t-1})$	0.0819	0.0199	<.001
$\log(t_i)$	-0.0471	0.0156	0.003
R^2=0.211, Adj R^2=0.193, DF=746, $p < 0.0001$			

Table 9: Hypothesis 3 – Multi-Level Regression Coefficients Predicting Number of Volunteer Developers by Project Classified By Firm Model (Equation 2)

Variable	Estimate	Std Err	P Value
$\log(VolDevs_{i,t-1})$	-0.4871	0.0356	<.001
$\log(VolDevs_{CF_{i,t-1}})$	0.0918	0.0350	0.008
$\log(VolDevs_{PF_{i,t-1}})$	-0.0212	0.0299	0.479
$\log(Commits_{i,t-1})$	0.0843	0.0194	<.001
$\log(t_i)$	-0.0401	0.0159	0.011
R^2=0.217, Adj R^2=0.198, DF=745, $p < 0.0001$			

Table 10: Hypothesis 4 – Testing for issues of cognitive complexity through the analysis of effect of commercial developers at the module level (Equation 3)

Variable	Estimate	Std Err	P Value
$\log(VolDevs_{i,j,t-1})$	-0.5405	0.0142	<.001
$\log(ComDevs_{i,j,t-1})$	0.0038	0.0132	0.774
$\log(Commits_{i,j,t-1})$	0.0992	0.0079	<.001
$\log(t_i)$	-0.0022	0.0055	0.693
$R^2 = 0.226$, Adj R^2=0.213, $DF = 5996$, $p < 0.0001$			

Table 11: Hypothesis 4 – Testing for issues of cognitive complexity through the analysis of effect of commercial developers at the module level (Equation 4)

Variable	Estimate	Std Err	P Value
$\log(VolDevs_{i,j,t-1})$	-0.5464	0.0142	<.001
$\log\left(ComDevs_{CF_{i,j,t-1}}\right)$	0.0616	0.0156	<.001
$\log\left(ComDevs_{PF_{i,j,t-1}}\right)$	-0.0284	0.0134	0.034
$\log(Commits_{i,j,t-1})$	0.0996	0.0075	<.001
$\log(t_i)$	0.0004	0.0055	0.934
$R^2 = 0.229$, Adj R^2=0.215, $DF = 5995$, $p < 0.0001$			

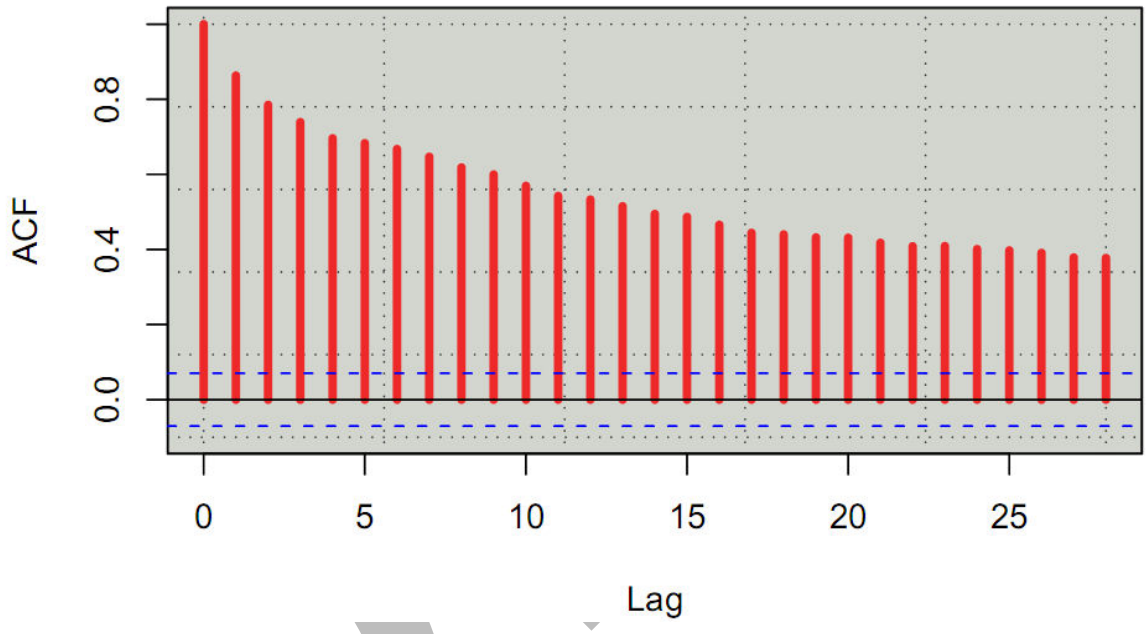

Figure 1: Autocorrelation between time periods of the number of volunteer developers in the GNOME community. One time unit represents eight weeks of calendar time.

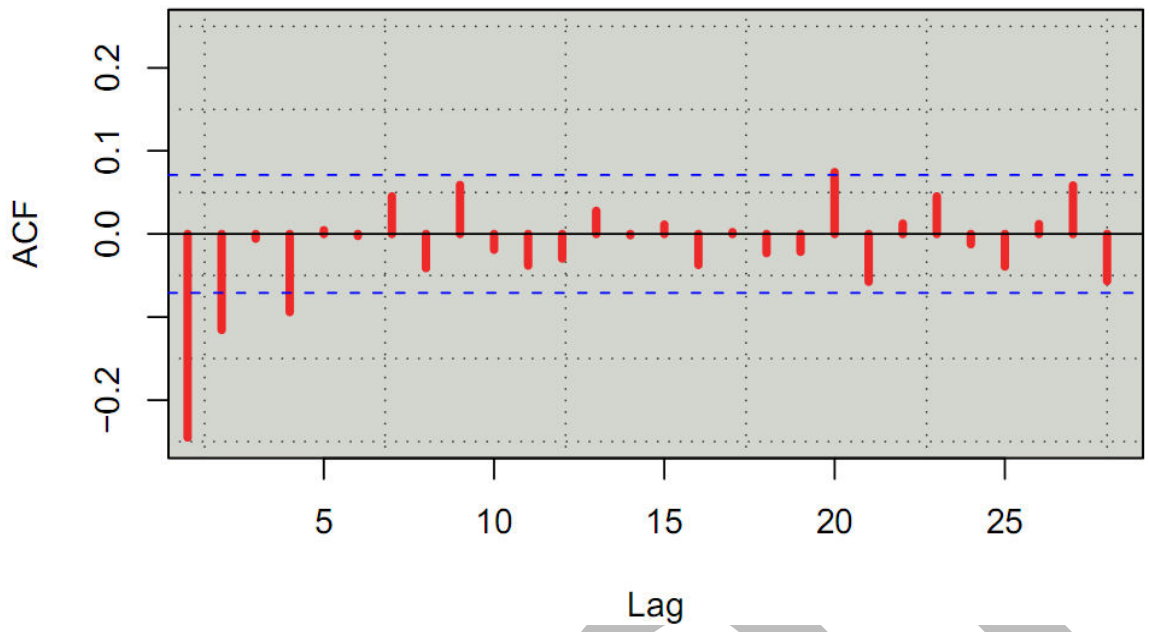

Figure 2: Autocorrelation between time periods of the change in the number of volunteer developers in the GNOME community. One time unit represents eight weeks of calendar time.