DISCUSSION PAPER SERIES

No. 6789

MICROSTRUCTURE OF COLLABORATION: THE ‘SOCIAL NETWORK’ OF OPEN SOURCE SOFTWARE

Chaim Fershtman and Neil Gandal

INDUSTRIAL ORGANIZATION

Centre for Economic Policy Research

www.cepr.org

Available online at: www.cepr.org/pubs/dps/DP6789.asp

ISSN 0265-8003

MICROSTRUCTURE OF COLLABORATION: THE 'SOCIAL NETWORK' OF OPEN SOURCE SOFTWARE

Chaim Fershtman, Tel Aviv University and CEPR
Neil Gandal, Tel Aviv University and CEPR

Discussion Paper No. 6789
April 2008

Centre for Economic Policy Research
90–98 Goswell Rd, London EC1V 7RR, UK
Tel: (44 20) 7878 2900, Fax: (44 20) 7878 2999
Email: cepr@cepr.org, Website: www.cepr.org

This Discussion Paper is issued under the auspices of the Centre's research programme in **INDUSTRIAL ORGANIZATION**. Any opinions expressed here are those of the author(s) and not those of the Centre for Economic Policy Research. Research disseminated by CEPR may include views on policy, but the Centre itself takes no institutional policy positions.

These Discussion Papers often represent preliminary or incomplete work, circulated to encourage discussion and comment. Citation and use of such a paper should take account of its provisional character.

CEPR Discussion Paper No. 6789

April 2008

ABSTRACT

Microstructure of Collaboration: The ‘Social Network’ of Open Source Software*

The open source model is a form of software development with source code that is typically made available to all interested parties. At the core of this process is a decentralized production process: open source software development is done by a network of unpaid software developers. Using data from Sourceforge.net, the largest repository of Open Source Software (OSS) projects and contributors on the Internet, we construct two related networks: A Project network and a Contributor network. Knowledge spillovers may be closely related to the structure of such networks, since contributors who work on several projects likely exchange information and knowledge. Defining the number of downloads as output we finds that (i) additional contributors are associated with an increase in output, but that additional contributors to projects in the giant component are associated with greater output gains than additional contributors to projects outside of the giant component; (ii) Betweenness centrality of the project is positively associated with the number of downloads. (iii) Closeness centrality of the project appears also to be positively associated with downloads, but the effect is not statistically significant over all specifications. (iv) Controlling for the correlation between these two measures of centrality (betweenness and closeness), the degree is not positively associated with the number of downloads. (v) The average closeness centrality of the contributors that participated in a project is positively correlated with the success of the project. These results suggest that there are positive spillovers of knowledge for projects occupying critical junctures in the information flow. When we define projects as connected if and only if they had at least two contributors in common, we again find that additional contributors are associated with an increase in output, and again find that this increase is much higher for projects with strong ties than other projects in the giant component.

JEL Classification: L17
Keywords: microstructure of collaboration, network and open source

Chaim Fershtman
The Eitan Berglas School of Economics
Tel-Aviv University
Tel Aviv 69978
ISRAEL
Email: fersht@post.tau.ac.il

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=104930

Neil Gandal
Department of Public Policy
Tel Aviv University
Tel Aviv 46766
ISRAEL
Email: gandal@post.tau.ac.il

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=144974

* We thank Rafi Aviav for very helpful research assistance. Research grants from Microsoft and the Net Institute are gratefully acknowledged. Any opinions expressed are those of the authors.

Submitted 01 April 2008

1. **Introduction**

The open source model is a form of software development with source code that is typically made available to all interested parties; users generally have the right to modify and extend the program.[1] The open source model has become quite popular and often referred to as a movement with an ideology and enthusiastic supporters.[2] At the core of this process is a decentralized production process: open source software development is done by a network of unpaid software developers. The developers typically work in different locations and yet contribute jointly to the projects in which they are involved. Since there are many such projects, these developers may be involved in more than one project and may work with different groups of co-developers in various open source projects.

Having unpaid volunteers is puzzling for economists. What are the incentives that drive developers to invest time and effort in developing these open source programs? There is a great deal of research on open source software and much of it focuses on the incentives to contribute to open source software projects. Lerner and Tirole (2002) argue that developers of open source programs acquire a reputation, which is eventually rewarded in the job market. Harhoff, Henkel and von Hippel (2003) argue that end users of open source benefit by sharing their innovations.[3] Using a Web-based survey Lakhani and Wolf (2005) find that intrinsic motivations help induce developers to contribute to OSS.[4]

Whenever co-workers collaborate on a joint project, they exchange information and create knowledge spillovers.[5] The phenomenon exists in commercial as well as in open source projects. When people interact, information is exchanged. Thus, the microstructure of the open source network might affect the R&D process and the spillovers of knowledge. When a network is relatively unconnected there will be less information flow between researchers. On the other hand, strongly connected networks imply relatively large flows among projects.

[1] Open source is different than “freeware” or “shareware.” Such software products are often available free of charge, but the source code is not distributed with the program and the user has no right to modify the program.

[2] See for example Raymond (2000) and Stallman (1999).

[3] Hann, Roberts, and Slaughter (2002) examine the Apache HTTP Server Project and find that contributions are not correlated with higher wages, but a higher ranking within the Apache Project is indeed positively correlated with higher wages. But such a correlation will occur whenever a higher ranking reflects higher productive capabilities of programmers.

[4] See also Hars and Ou (2001), Hertel, Niedner, and Herrmann (2002). Using survey methods, these papers respectively find that peer recognition and identification with the goals of the project are the main motivations for developers who contribute to open source software projects.

[5] For a model of an R&D race with spillovers see D'Aspermont and Jacquemin (1988).

Goyal and Moraga (2001) for example examine the interaction between the architecture of the collaboration network and the firm incentives to invest in R&D.

There is a large economics literature that examines the properties of social networks, their formation and the relevant economic implications. (For surveys see Jackson (2006,2008) and Goyal (2007) and for general methods and applications see Faust and Wasserman (1994)). The focus of this literature is mainly on network formation, strategic interaction in networks and the effect of network structure on behavior. While our paper is more related to the literature on 'the effect of network structure on behavior' than the other literatures, the focus of our paper is quite different: it is on the relationship between network properties and output/success of different nodes (projects) in this network.[6]

In this paper, we study the structure of the open source network. We use the data from Sourceforge.net, which is the largest repository of OSS code and applications available on the Internet, with 114,751 projects and 160,104 contributors.[7] We primarily focus on the relationship between the network structure and the success of open source projects. Each SourceForge project page links to a "Developers page" that contains a list of registered team members.[8] The Sourceforge.net information structure is rooted in projects. The data from SourceForge.net form a two-mode-network of projects and contributors. Using these data, we can construct the project network in the following way: there is a link between two projects if there is at least one contributor who works on both projects. Similarly, we can construct the contributor network, such that there is a link between two contributors if they work on at least one project in common.[9] One can also construct a weighted network such that the weight of each link between two contributors is the number of projects that two contributors jointly participated and similarly the weight of a link between two projects is the number of contributors that participated in both.

Interestingly, both the project network and the contributor network consist of one "giant" connected component and many smaller unconnected networks: in the case of the project

[6] For the effect of network structure on behavior, see for example Ballester, Calvo-Armengol and Zenou (2006) and Goeree, McConnell, Mitchell, Tromp and Yariv (2007).

[7] These numbers are from June 2006 when we collected our data.

[8] Sourceforge.net facilitates collaboration of software developers, designers and other contributors by providing a free of charge centralized resource for managing projects, communications and code.

[9] The construction of the contributor network is similar to the construction of the coauthor network in Economics by Goyal, van der Leij and Moraga (2006). Our emphasis however is not so much on the properties of this network but on the relationship between these properties and the success of different projects.

network, the giant component contains 27,246 connected projects, while the second largest connected component consists of only 27 projects. In the case of the contributor network, there is a giant component of 55,087 connected contributors and many smaller components.[10] 77% of the contributors worked only on a single project while at the other end of the spectrum, there are a small number of "stars" who work on many projects (there are 344 contributors that worked on ten or more projects).

It is not easy to measure the success of open source software. Like other products based on intellectual property, the intellectual property in software (including open source software) is "licensed" for use. In the case of commercial software, however, there are license fees; thus it is possible to determine the number of licenses issued, as well as the revenues earned from these licenses. That is not the case with open source software, which does not have license fees and information on the number of licenses is not available. One way to measure project success is to examine the number of times a project has been downloaded. Clearly, this is not an ideal measure. Nevertheless, downloads are often used in order to measure the impact of academic papers and articles on the web.[11] Hence, we assume that the number of downloads of open source projects is likely quite correlated with use and value.[12]

In addition to downloads, there are three groups of variables that we use in the analysis. The first is a group of control variables that includes the amount of time that the project has been in existence, the stage of development, the number of operating systems for which the program was written, the number of languages in which the program is written, as well as several other control variables. We also employ a group of network variables, which can be further broken down to two subgroups. The first group includes variables (like *degree* – the number of links) that are comparable across all projects. The second group of network variables includes two network centrality measures; these variables are only comparable for projects in linked components. The *betweenness centrality*, or *betweenness*, of a node is defined as the proportion of all geodesics between pairs of other nodes that include this node, where a geodesic is the shortest path between two nodes. *Betweenness* captures the notion that a node is considered central if it serves as a valuable juncture between other nodes. The

[10] The second largest component in the contributor network consists of only 196 contributors.

[11] The Social Science Research Network, for example, provides information on the number of downloads for the papers on its website.

[12] We will also show that in the case of the Sorceforge.net data, the number of project downloads is especially large for projects selected "project of the month" at SourceForge. This reinforces the notion that downloads is a good measure of success.

closeness centrality, or *closeness*, of a node is defined as the inverse of the sum of all distances between the node and all other nodes, multiplied by the number of other nodes. C*loseness* measures how far each project is from the other projects in the network.

Our first result is that additional contributors are associated with higher output (downloads), both for projects in the giant component and projects outside of the giant component, but the increase in downloads associated with an increase in contributors is much larger for projects in the giant component. This robust result obtains even though the average number of contributors is higher for projects in the giant component.

We then examine how the network centrality measures affect the number of downloads. Since these network centrality measures are only comparable across connected components, we conduct this analysis for projects in the giant component. We find that *betweenness centrality* is highly associated with a higher number of downloads. Since projects with higher values of *betweenness* are positioned in heavier information flows, our results suggest that projects "well-positioned" in information flows are more successful and there are positive spillovers of knowledge for projects occupying critical junctures in the information flow.

Closeness centrality appears also to be positively associated with downloads, but the effect is not statistically significant over all specifications. Controlling for the correlation between these two measures of centrality (*betweenness* and *closeness*), *degree* is not positively associated with the number of downloads.

We are careful not to attach a causal interpretation to our results because it is not possible to determine from the data whether increases in network measures (e.g. number of contributors, *betweenness* or *closeness*) increase downloads or whether highly successful projects attract more productive contributors. Although the data do not afford an opportunity to investigate causality, we document the ways in which projects with more downloads differ from projects with fewer downloads. We believe that the results are interesting because they show which network and centrality measures are most highly correlated with success.

Throughout most of the paper, we define projects as connected if and only if they had at least one contributor in common, that is, we ignored the weight of the link. An interesting question to ask is whether the strength of the links has any effect on the success of the projects. When we define projects as connected if and only if they had at least two contributors in common, the largest component of strongly connected projects consists of only 259 projects. We find that additional contributors are associated with an increase in output, and that this increase is 150% greater for projects in the component with stronger ties, than other projects in the giant component.

Finally, we turn to examining the contributor network and its possible effect on the projects' success. After controlling for the correlation of the "project" characteristics with "project" success we find that the average closeness centrality of the contributors that participated in a project is positively correlated with the success of the project..

2. The Two-Mode Network of Contributors and Projects

We obtained our data by “spidering” the website http://SourceFourge.net, which is the largest Open Source software (OSS) development web site.[13] The data was retrieved from SourceForge.net during June 2006 and includes 114,751 projects and 160,104 contributors who were listed in these projects. The contributors are identified by unique user names they chose when they registered as members in SourceForge. The site’s information structure is rooted in projects. The interface of SourceForge.net allows almost all of the information about the projects to be viewed by anyone.[14] Each project has a “Project page” which is a standardized ‘home page’ that links to all the services and information made available by SourceForge.net for that project. The project page itself contains important descriptive information about the project, such as a statement of purpose, the intended audience, license, operating system etc.

Each project page links to a “Statistics page” that shows various activity measures, such as the number of downloads. Each project page also links to a “Developers page” that has a list of registered team members. This list is managed by the project administrators who are also listed as team members. The assumption in this paper is that the site members who are listed as project team members were added to the list because they made a contribution to the project that involved investment of time and effort. A project is thus seen as a collaborative effort by its team members, or *contributors*.

[13] Spidering is term used to describe recursive algorithms used to traverse a website page-by-page and automatically extract desired information based on forms and content pattern.

[14] A very small number of projects block certain data from being accessed by anyone who isn’t a project team member.

The data we obtained from SourceForge.net form a two-mode-network of projects and contributors. A two-mode-network is a network partitioned into two types of nodes, e.g. projects and contributors. We can use the two-mode network to construct two different one-mode networks: (i) the contributors' network and (ii) project network.

Contributor Network:

- The nodes of this network are the contributors, i.e., the distinct names (or emails) of the contributors.
- There is a link between two different contributor nodes if the two contributors participated in at least one OSS project together.
- Each link may have a value which reflects the number of projects in which the contributors jointly contributed.

Projects Network:

- The nodes of this network are the OSS projects.
- There is a link between two different project nodes if there are contributors who participate in both projects.
- Each link may have a value which reflects the number of contributors that participate in both projects.

The following table shows the distribution of contributors per project and projects per contributor for the two-mode-network at Sourceforge.net.

Project network		Contributor network	
Contributors per project	Number of projects	Projects per contributor	Number of contributors
1	77,571	1	123,562
2	17,576	2	22,690
3-4	11,362	3-4	10,347
5-9	6,136	5-9	3,161
10-19	1,638	10-19	317
20-49	412	20-49	26
≥50	56	≥50	1
Total Projects	**114,751**	Total Contributors	**160,104**

Table 1: The distribution of contributors per project and projects per contributor

Table 1 shows that 68% of the projects hosted at Sourceforge.net have just a single contributor.[15] An additional 15% of the projects have two contributors. At the other end of the spectrum, there are 1,638 projects with 10-19 contributors and 468 projects with more twenty or more contributors. Similarly, Table 1 shows that 77% of the contributors worked on a single project, while an additional 14% contributed only to two projects. Thus more than 90% of the open source contributors worked on just one or two projects. At the other end of the spectrum, there are a small number of "stars" who work on many projects: 3,161 contributors worked on 5-9 projects, while 344 contributors worked on ten or more projects.

There are six levels of development that range from the planning stage to a mature status. There is an additional status reserved for projects that are inactive. Table 2 below provides the distribution of the development status for the single contributor and the multi-contributor projects. As is evident from this table the two distributions are similar. The possibility that the single contributor projects are in some way infant projects thus seems remote. In any case, we will control for the time for which the project has been in existence.

[15] While these projects do not provide links between contributors, such contributors who work on multiple projects provide links among projects.

Development status	Relative frequency in "single contributor" projects	Relative frequency in "multi contributor" projects
1 – Planning	21%	21%
2 - Pre-Alpha	17%	16%
3 – Alpha	18%	17%
4 – Beta	22%	23%
5 – Production/Stable	18%	20%
6 – Mature	1%	2%
Inactive	2%	2%

Table 2: Development Status

2.1 The Network of Contributors:

For the contributor network, there is a link between contributors *i* and *j* if they have worked on at least one project in common. The set of contributors can be divided into components such that all of the contributors in a component are connected to one another and there is no sequence of links among contributors in different components. The distribution of the components is shown in Table 3a. There is a "giant" component, which consists of 55,087 contributors, or approximately 45% of the contributor network. The table shows that there are many small components as well.

Component size (Contributors)	Components (sub networks)
55,087	1
196	1
65-128	2
33-64	27
17-32	152
9-16	657
5-8	2,092
3-4	4,810
2	8,287
1	47,787

Table 3a: Distribution of component size

Degree	Number of contributors
0	47,787
1	22,133
2	14,818
3-4	20,271
5-8	20,121
9-16	16,228
17-32	10,004
33-64	5,409
65-128	2,040
129-256	802
257-*505*	491

Table 3b: Distribution of Degree

For every contributor in the network, we can define the degree as the number of links between that contributor and other contributors in the network.[16] Table 3b shows the distribution of degree in the contributor network. There are 47,787 contributors who work only in single contributor projects. At the other end of the spectrum 491 "star" contributors worked on projects in common with more than 256 other contributors.

2.2 The Network of Projects:

In the project network, a node is a project and there is a link between two projects if and only if there are contributors who have contributed to both projects. Table 4a shows that the project network consists of one “giant” connected component with 27,246 projects and many smaller unconnected components. The giant component contains approximately 24% of the projects at the Sourceforge website. It is indeed striking that the second largest “network” consists of only 27 projects. The *degree* of a project is the number of other projects with which that project has a link. Table 4b shows the distribution of *degree* for the project network. Two-thirds of the project have degree less than or equal to one. At the other end of the spectrum, 370 projects have degree greater than thirty-two.

Size	Connected components
27,246	1
17-*27*	36
9-16	234
5-8	1,013
3-4	3,419
2	8,020
1	51,093

Table 4a: Distribution of component size

Degree	Number of projects
0	51,093
1	22,926
2	12,709
3-8	22,004
9-32	5,649
33-64	290
≥65	80

Table 4b: Distribution of degree

2.3 Measuring Success/Output in the Project Network

Defining or measuring the success of an open source project is problematic. There are no prices and no ‘sales’. The projects are in the public domain and there is no need to provide

[16] Hence, a contributor who worked on a single project with four other contributors has a degree of four. Similarly, a contributor who worked on two projects, each of which had two additional contributors (who only worked on one of the two projects), would also have a contributor degree equal to four.

payment or request permission in order to use them. One way to measure project success is to examine the number of times a project has been downloaded. Clearly, this is not an ideal measure, as there is a difference between downloads and usage or value. However, downloads are also often used in order to measure the impact of academic papers and articles on the web.[17] The Social Science Research Network, for example, provides information on the number of downloads for the papers on its website. We assume that the number of downloads of open source projects is a proxy for the use and value of the OSS projects.

Every month, the Sourceforge.net staff chooses a “project of the month.” Although we do not know the exact criteria that are employed in choosing the “project of the month,” these projects are likely to be very “successful.” We obtained data on the "project of the month" for the forty-two month period ending in June 2006. The “project of the month” projects have an especially large number of downloads.[18] “Project of the month” projects are typically in advanced stages (stages 4,5, and 6); thirty-eight of the forty-two projects of the month projects are either in stage 4, stage 5, or stage 6. The thirty-eight “project of the month” projects in advance stages had on average 6,028,560 downloads, versus 30,206 downloads (on average) for the other 35,821 projects in advanced stages. The median number of downloads for “project of the month” projects in advance stages was 1,154,469 versus 483 for other projects in advance stages. This suggests that the number of project downloads is an attractive measure of use and value.

There are several different download measures that we could use: (i) the total number of downloads since the project was initiated at Sourceforge.net (ii) the maximum number of downloads in any month, and (iii) the number of recent downloads. The correlation among these download measures is, however, quite high. Since it contains the most information, we chose to use the total number of downloads in our analysis. Henceforth, when we refer to downloads, we mean the total number of downloads and denote *downloads* as the total number of downloads for the forty-two month period for which we have data. We further define *ldownloads* ≡ ln(1+*downloads*), where “ln” means the natural logarithm. Since it may take some time for projects to reach an “equilibrium” level of contributors, we will also

[17] Indeed, in a way academic papers are like open source projects. Typically, no permission or licensing agreement is required to access an academic publication.

[18] Given that there are only forty-two such “projects of the month,” we cannot use this as our measure of success.

perform robustness checks by conducting the analysis for projects that have been in existence for at least two years.

3. Data and Variables Available for the Analysis

In addition to downloads, there are three groups of variables that we use in the analysis. The first is a group of control variables that includes the amount of time that the project has been in existence, the stage of development, the number of operating systems for which the program was written, the number of languages in which the program is written, as well as several other control variables. We also employ a group of network variables, which can be further broken down to two subgroups. The first group includes variables (like *degree*) that are comparable across all projects, regardless of whether the projects are linked. The second group of network variables includes *betweenness* and *closeness*; these variables are only comparable for projects in linked components. When we use the last set of variables, we will restrict the analysis to the giant component. The variables are as follows:

<u>(i) Control Variables:</u>

- The variable *years_since* is the number of years that have elapsed since the project first appeared at Sourceforge: *lyears_since*=ln(*years_since*).

- The dummy variable *ds_j* refers to the stage where j ranges from one to six. There is an additional stage, denoted *inactive*, which means the project is no longer active. See Table 2. A few of the projects are considered to be in multiple stages. Hence, for a particular project, it is possible that both ds_3 and ds_4 could be equal to one.

- The variable *count_trans* is the number of languages in which the project appears including English. Virtually all of the projects (95%) are available in English. The other popular languages include German (5% of the projects), French (4%), and Spanish (3%). *lcount_trans*=ln(*count_trans*)

- The variable *count_op_sy* is the number of operating systems (i.e., formats) in which the project is compatible. Some of the projects are available for several operating systems. The main operating systems in which the projects were written include Windows (32% of the projects), Posix (26% of the Projects), and Linux (21% of the Projects. *lcount_op_sy*=ln(*count_op_sy*)

- The variable *count_topics* is the number of topics included in the project description. Popular topics include the Internet (16% of the projects), software development (14%), communications software (11%), and games & entertainment software (10%). *lcount_topics*=ln(*count_topics*)

- The variable *count_aud* is the number of main audiences for which the project was intended. The main audiences are developers (35% of the projects), end users (30% of the projects), and system administrators (13% of the projects). Some of the products are intended for multiple 'main audiences' while other projects are not intended for these main audiences, but rather just for niche audiences, i.e., just for a particular industry (i.e., telecommunications) or just for very sophisticated end users. *lcount_aud*=ln(1+*count_aud*)

Clearly, there are different ways to include variables on translations, operating systems, topics and audiences. For example, we could have simply counted the key operating systems, or used dummy variables for these operating systems. Similarly, we could have defined dummy variables for 'main audiences' or we could have added up the number of main audiences together with the number of niche audiences. We chose the definitions that seemed most natural. The main results regarding the number of contributors and the network variables are robust to alternative definitions of these control variables.

(ii) Network Variables defined for all projects:

- The variable *cpp* is the number of contributors that participated in the project: *lcpp*=ln(*cpp*)

- *degree* - The degree for a project is the total number of projects, with which it has at least one contributor in common. *ldegree*=ln(1+*degree*)

- *giant_comp* is a dummy variable that takes on the value one if the project is in the giant component, and takes on the value zero otherwise.

In order to allow for the possibility that the association between *degree* and downloads and between the number of contributors and downloads depends of whether the project is inside or outside of the giant component, we also include the following interaction variables in the analysis:

- *lgiant_degree* = ldegree*giant_comp,
- *lgiant_cpp* = lcpp*giant_comp,

Descriptive statistics in Table A1 of the appendix show that, not surprisingly, the mean *degree* and the number of contributors are higher for projects in the giant component. By including the interaction variables, we allow for the possibility that there will be different download "elasticities" for projects in and projects outside of the giant component.[19]

[19] The addition of different slopes for the control variables based on whether the project was inside or outside of the giant component has no effect on the main results regarding the number of contributors and the degree of the project.

(iii) Network Variables that are comparable only among linked projects:

It is postulated that the "importance" of nodes in a network depends on their centrality. Hence, we introduce two key measures of centrality that are typically used in social network theory: *betweenness centrality*, and *closeness centrality*. For a network of size "#N," the *betweenness centrality*, or *betweenness*, of a node is defined as the proportion of all geodesics between pairs of other nodes that include this node, where a geodesic is the shortest path between two nodes. Formally[20], the *betweenness* of a node i is given by

$$(1) \qquad C_B(i) \equiv \frac{\sum_{\substack{j<k \\ i \notin \{j,k\} \subseteq N}} \left[\gamma_{jk}(i) / \gamma_{jk} \right]}{(\#N-1)(\#N-2)/2}$$

where γ_{jk} is the number of distinct geodesics between the nodes j and k which are distinct from i, and $\gamma_{jk}(i)$ is the number of such geodesics which include i.[21] *Betweenness* captures the notion that a node is considered central if it serves as a valuable juncture between other nodes. We further define *lbetween*=ln(.0001+*betweenness*)[22]

For any two nodes $i, j \in N$, the distance or degree of separation between them (denoted $d(i, j)$) is the length of the geodesic between them. *Closeness centrality* of a node is defined as the inverse of the sum of all distances between the node and all other nodes, multiplied by the number of other nodes, so that it lies in the range [0,1].[23] Formally, *closeness centrality* is calculated as follows:

$$(2) \qquad C_C(i) \equiv \frac{\#N-1}{\sum_{j \in N} d(i,j)}$$

C*loseness* measures how far each project is (on the average) from the other projects in the network. We further define *lcloseness*=ln(0.05+*closeness*).[24]

[20] See Freeman (1979) for quantification of this notion.

[21] The denominator of (1) is the maximum possible value for the numerator, and thus standardizes the measure in the range [0, 1].

[22] The reason we add such a small number is because the mean value of *betweenness* is 0.00028.

[23] See Faust and Wasserman (2005), p 184-185.

[24] The reason we add such a small number is because the mean value of *closeness* is 0.14.

4. Analysis: Characteristics Associated with the Success of Projects

In this section, we examine the relationship between downloads and the control and network variables. We estimate a simple log/log model of the form $ldownloads_i = \alpha + \beta N_i + \gamma C_i + \varepsilon_i$, where the subscript *i* refers to the project. N_i is the natural logarithm of the "network variables" and C_i is the natural logarithm of the control variables. For binary ([0,1]) variables, we, of course do not employ logarithms; ε_i is a random error term.

We have data on 114,450 observations for all of the network variables as well as on *years_since*.[25] However, data on the stage of development and the count variables are incomplete; data on all of the control variables are available only for 66,511 projects. Since there is no selection issue,[26] we do what is typically done in such cases and use only the data on the 66,511 projects for which we have complete information.[27] In section 4.1, we conduct an analysis using these projects and examine the association between the control and network variables and success. In section, 4.2 we follow up this analysis by examining the giant component in detail (18,697 projects for which there is complete information), which enables us to include the variables *betweenness* and *closeness* in the analysis. We then perform robustness checks by examining established projects only (section 4.3) and projects with more than one contributor (section 4.4).

Descriptive statistics of the variables are shown in Table A1 in the appendix. Table A1 shows that projects in the giant component have on average more downloads than projects outside of the giant component (42,751 vs. 10,959). Further, projects in the giant component are on average (i) older than projects outside of the giant component (3.63 years vs. 2.70 years), (ii) have more contributors (3.84 vs. 1.61), and have a larger degree (6.26 vs. 1.18).[28]

[25] There are 114,751 total projects, but we are missing data on downloads for a small number of them (301).

[26] See Griliches (1986) and Green (1993).

[27] We do not discard the information that these projects provide concerning the network structure and the values of network variables that are included in the database. Further, it is comforting to know that our main results regarding the association between the number of contributors and success and the centrality variables and success are qualitatively unaffected by whether we use the full data set, or the observations for which we have data on all relevant variables. These results are available from the authors on request.

[28] Correlations among the independent variables in the regressions are shown in Table A2 in the appendix.

4.1 Analysis Using All Projects

The results of a regression with all 66,511 observations are shown in the first column of Table 5.[29] The estimated coefficients show that the association between downloads and the number of contributors is positive – projects with more contributors have greater downloads. For projects outside of the giant component, the estimated "contributor" elasticity is 0.46. That is, a one percent increase in the number of contributors is associated with a 0.46 percent increase in the number of downloads. This effect is statistically significant. The estimated "contributor" elasticity is virtually twice as large for projects in the giant coefficient: 0.90 (0.46+0.44). The difference in the estimated "contributor" elasticity between projects in the giant component and projects outside of the giant component is statistically significant: additional contributors are associated with greater increases in output for projects in the connected (giant) component than in the non-connected component. This result obtains despite the fact that there are many more contributors (on average) for projects in the giant component (3.84 vs. 1.61). One possible explanation for this result is that the contributors to projects in the giant component are more skilled than the contributors who work on projects outside of the giant component. Alternatively, it could mean that there are knowledge spillovers among projects with ties that enhance the productivity of those who work together on these projects.

The *degree* elasticity, i.e., the association between *degree* of the project and the number of downloads, is positive and statistically significant both for projects inside the giant component and for projects outside of the giant component. This suggests that projects with a higher *degree* are associated with higher output. For projects outside of the giant component, the *degree* elasticity is 0.19, while the *degree* elasticity for projects in the giant component is 0.14. Both of these magnitudes are statistically significant from zero; the difference in the magnitudes is not statistically significant.

The estimated coefficient of *lyears_since* is positive (1.42) and statistically significant. This suggests that projects that have been active longer have more downloads, and the estimated coefficient suggests that a doubling of the time a project has been active is associated with 142% more downloads.[30] The estimated coefficients on the stage variables have the expected

[29] Correlations among the independent variables in the regressions are shown in Table A2 in the appendix.

[30] In section 4.3, we show that our results regarding the association between the number of contributors and downloads and the association between the centrality measures *(betweenness and closeness*) and downloads is robust to excluding projects that are less than two years old.

signs. By and large, projects that are in more advanced stages are associated with more downloads. Similarly, projects written for several operating systems, projects available in more languages, projects written for more main audiences, and projects that span more topics are associated with more downloads as well.

Dept Variable: ldownloads	Regression 1 (All 66,511 Projects)		Regression 2 (Giant Component - 18.697 Projects)	
Independent Variables	Coeff.	T-stat	Coeff.	T-stat
Constant	0.72	17.76	5.71	10.76
lyears_since	1.42	60.66	1.68	31.14
lcount_topics	0.23	9.07	0.18	3.66
lcount_trans	0.35	11.73	0.43	7.85
lcount_aud	0.36	10.44	0.41	5.52
lcount_op_sy	0.11	5.95	0.18	4.92
ds_1	-1.96	-60.57	-2.02	-32.24
ds_2	-0.60	-17.58	-0.80	-11.89
ds_3	0.89	25.83	0.64	9.76
ds_4	1.86	57.21	1.78	29.08
ds_5	2.72	79.97	2.58	40.65
ds_6	2.12	27.07	2.01	15.31
inactive	0.45	6.11	0.35	2.54
Lcpp	0.46	18.71	0.61	16.71
ldegree	0.19	9.45	-0.13	-3.12
Giant_comp	-0.21	-3.86		
lgiant_cpp	0.44	12. 05		
lgiant_degree	-0.05	-1.26		
betweenness			0.48	12.15
closeness			0.38	1.76
# of Observations	66,511		18,697	
Adjusted R-squared	0.41		0.41	

Table 5: Regression Results: Dependent Variable: ldownloads

4.2 Analysis for the Giant (Connected) Component

We now turn to discuss the relationship between downloads and the two centrality measures that we defined above: *betweenness* and *closeness*. In the second regression in Table 5, we add these centrality variables to the analysis. Since *betweenness* and *closeness* are only comparable across linked networks, this regression is done for the giant component only. The results from this regression suggest that the contributor elasticity (0.61) is again statistically significant.

The estimated *betweenness* elasticity (0.48) is positive and statistically significant. Thus, projects that sit in critical information flows have greater downloads. The estimated *closeness* elasticity (0.38) is statistically significant as well at the 0.92 level: projects that are relatively 'close' to other projects have more downloads. These results suggest that it is not just the ties among projects (via contributors) that matter for downloads, but how the projects are tied together and their position in the network.

The estimated degree elasticity is negative (-0.13) in this regression. This suggests that controlling for *betweenness* and *closeness* centrality, there is not a positive association between the number of downloads and the *degree* of the project: the two other centrality measures are more important for the number of downloads than is the degree of the project.

The estimated coefficient of *lyears_since* is again positive (1.68) and statistically significant. The estimated coefficients on the stage and count variables again have the expected signs and are qualitatively similar to those in the first regression in Table 5.

We now define a "star" as a contributor who worked on five or more projects. An interesting question is if having a "star" in the team of developers has an effect on the success of a project. To examine this, we add a dummy variable (denoted star) -- which takes on the value one if the project has at least one star and takes on the value zero otherwise -- to the second regression in Table 5.[31] We find that although the effect is not statistically significant (coefficient=0.10, t=1.41), the presence of a "star" contributor is positively correlated with the success of the project. This effect, which obtains even after controlling for measures of project centrality, suggests that star contributors are associated with positive information spillovers beyond what is accounted for by the centrality measures. The estimated coefficients on betweenness and closeness are unaffected by the addition of "star."

4.3 Robustness of Results to Inclusion of Established Projects Only

Nascent projects may not have reached a steady-state number of contributors. Personnel additions are probably more likely for relatively new products. It is important to know whether the results are robust to using only established projects in the analysis. Hence, we re-did the regressions in Table 5 for projects that had been in existence for at least two years.[32] Our results are qualitatively unchanged.

[31] 92% of the projects outside of the giant component do not have a star. 45% of the projects in the giant component have at least one star.

[32] The median age of projects in our data set as of June 2006 was 2.66 years.

In the case of the first regression, we are left with 44,638 observations (or 67% of the observations) when we restrict the analysis to projects that had been in existence for more than two years. For projects outside of the giant component, the estimated “contributor” elasticity is 0.51 (versus 0.46 in Regression 1 in Table 5), while the estimated “contributor” elasticity for projects in the giant coefficient is 0.91 (virtually the same as in the first regression in Table 5). The difference in the estimated “contributor” elasticity between projects in the giant component and projects outside of the giant component is again statistically significant.

The estimated coefficients on *degree* for projects outside and inside the giant component are positive and statistically significant (0.21 and 0.14 respectively), nearly the same as in the first regression in Table 5. The estimated coefficients on the stage and count variables again have the expected signs.

When we run a regression analogous to the second regression in Table 5 for projects in the giant component that have been in existence for more than two years, we are left with 14,749 projects (or nearly 79% of the observations). The estimated contributor elasticity (0.63 in this new regression versus 0.61 in the second regression in Table 5) is again positive, statistically significant and virtually unchanged. The estimated *betweenness* elasticity (0.47 in this new regression versus 0.48 in the second regression in Table 5) is again positive, statistically significant and virtually unchanged. The estimated *closeness* elasticity (0.31 in this new regression versus 0.38 in the second regression in Table 5) is not statistically significant (t=1.24). The estimated *degree* elasticity is -0.11 (-0.13 in the second regression in Table 5) is virtually unchanged. (For ease of presentation, these two regressions appear in the Appendix in Table A3.)

4.4 Robustness of results to projects with more than one contributor

In this section, we repeat the analysis for projects with more than one contributor. We are left with 25,422 projects when we restrict the analysis to projects that have more than one contributor. For projects outside of the giant component, the estimated “contributor” elasticity is 0.46 (virtually the same as in Regression 1 in Table 5), while the estimated “contributor” elasticity for projects in the giant coefficient is 1.01 (versus 0.90 for the same as in the first regression in Table 5). The difference in the estimated “contributor” elasticity between projects in the giant component and projects outside of the giant component is again statistically significant.

The estimated coefficients on degree for projects outside and inside the giant component are positive and statistically significant (0.15 and 0.18 respectively), and quite similar to the first regression in Table 5. The estimated coefficients on the stage and count variables again have the expected signs.

When we run a regression analogous to the second regression in Table 5 for projects in the giant component with more than one contributor, we are left with 11,814 projects with more than one contributor. The estimated contributor elasticity (0.75 in this new regression versus 0.61 in the second regression in Table 5) is again positive and statistically significant. The estimated *betweenness* elasticity (0.45 in this new regression versus 0.46 in the second regression in Table 5) is again positive, statistically significant and virtually unchanged. The estimated *closeness* elasticity is 0.44 in this new regression versus 0.38 in the second regression in Table 5. This coefficient (with a t-value of 1.53) is not statistically significant at the 0.90 level. The estimated *degree* elasticity, -0.14, is virtually unchanged from the second regression in Table 5.

The robustness analysis in sections 4.3 and 4.4 reinforce our main results: (i) the association between the number of contributors and the number of downloads is higher for projects inside the giant component than it is for projects outside of the giant component and (ii) *Betweenness* centrality is the centrality measure most highly associated with the number of downloads. *Closeness* centrality appears also to be positively associated with downloads, but the effect is not statistically significant over all specifications. Controlling for the correlation between these two measures of centrality (*betweenness* and *closeness*), *degree* is not positively associated with the number of downloads. (For ease of presentation, these two regressions appear in the Appendix in Table A4.)

5. The Importance of Strong Ties

So far we defined two projects to be linked if there was at least one contributor in common between them. But the potential information flow, or spillovers, between projects may depend also on the number of contributors that participated in the two projects. To capture this effect in this section we change the definition of a link and focus on "strong" links. Two projects are ‘strongly’ linked if and only if they have at least two contributors in common. That is, we define a new network in which the nodes are still projects, but the links are only 'strong' links.

Redefining the network has a dramatic effect on it structure. Previously in a network in which one contributor in common was sufficient for a link, there was a giant component of 27,246 projects. In the new network, the largest component of strongly connected projects consists of only 259 projects. There are four smaller strongly connected components with between 50-75 projects. Figure A1 in the Appendix shows the network structure of the largest component in the "strongly connected" network. A comparison of the median number of downloads between projects in the strongly connected component and other projects in the giant component suggest that a stronger connection is associated with more downloads. See Table 6.[33]

Group	# of projects	Mean # downloads	Median # downloads
Strongly Connected Component	259	82,238	2,035
Other Projects in Giant Comp.	26,897	30,230	98

Table 6: Strongly Connected Component vs. Other Projects in Giant Component.

We then run a regression employing three additional variables:

(i) A dummy variable for projects in the strongly connected component, denoted *strong*.

(ii) (ii) The variable *l*strong*_degree = ldegree* strong*.

(iii) (iii) The variable *l*strong*_cpp = lcpp* strong*.

We again find that additional contributors are associated with an increase in output, but that this increase is much higher for projects in the strongly connected component, than other projects in the giant component. The estimates of the contributor elasticity are 0.61 for projects in the giant component that are not part of the strongly connected component and 1.58 for projects that are in the strongly connected component (see Table 7). This suggests that strong ties make a large difference in the contributor elasticity. The other results are (not surprisingly) virtually unchanged from the second regression in Table 5.

[33] The same qualitative result obtains if we restrict the analysis to projects in stages 4-6. In this case, the projects in the strongly connected component have a median of 11,230, while other projects in the giant component in the same stages have a median of 1,431.

Dept Variable: ldownloads	Giant Component Projects with data on stage & count variables	
Independent Variables		
Constant	5.65	10.63
lyears_since	1.68	31.17
lcount_topics	0.18	3.63
lcount_trans	0.43	7.92
lcount_aud	0.41	5.53
lcount_op_sy	0.18	4.95
ds_1	-2.02	-32.25
ds_2	-0.80	-11.89
ds_3	0.64	9.74
ds_4	1.78	29.07
ds_5	2.58	40.61
ds_6	2.02	15.32
Inactive	0.36	2.54
lcpp	0.61	16.50
Ldegree	-0.13	-3.09
strong_comp	-0.19	-0.23
lstrong_cpp	0.97	3.00
lstrong_degree	-0.56	-1.64
Betweenness	0.47	11.95
Closeness	0.38	1.74
# of Observations	18,697	
Adjusted R-squared	0.41	

Table 7: Regression Results Adding Variables for Largest Strongly Connected Component

6. Contributor Characteristics Associated with Project Success

Up until this point, we focused on how project characteristics were associated with the success of the projects. We now add information regarding the contributors' network. In particular we know which contributors participated in each project and the network characteristics of these contributors. Our focus is to examine whether – after controlling for the correlation of project characteristics with project success – centrality measures of the contributor network are correlated with project success. In order to examine this issue, we created three new variables:

(i) Average *degree* of the contributors on a project.

(ii) The average *betweenness centrality* of the contributors to a project.

(iii) The average *closeness centrality* of the contributors to a project.

These variables differ respectively from the degree of a project, the betweenness centrality of a project and the closeness centrality a project. For example, consider a project (denoted A) with two contributors (denoted I and II), each of whom works on one other project. This means that project A has a (project) degree equal to two. Further suppose that contributor "I" also works on project B, and that there are three other distinct contributors on project B. Similarly, suppose that contributor II also works on project C, and that there are again three additional distinct contributors on project C. The "contributor" degree of contributor I equals four (since he/she participates with four other contributors in two different open source projects). Similarly, the contributor degree of "II" is four as well. Hence, the average contributor degree of project A is four. The average betweenness centrality of the contributors to a project and the average closeness centrality of the contributors to a project are analogously defined. While the degree of the project and the average degree of the contributors on a project are relatively highly correlated (0.64),[34] there is virtually no correlation (i) between the *closeness centrality* of a project and the average *closeness centrality* of its contributors (0.03) and (ii) between the *between centrality* of a project and the average *betweenness centrality* of its contributors (0.02).

When we add these three contributor network variables to regression 2 in Table 5, we find that controlling for the project factors, the average *closeness centrality* of the contributors who participate in the project is positively correlated with the success of the project. The t-value of the coefficient associated with this variable is t=1.72 if we include the other two "contributor" variables in the regression[35] and t=1.26 if we do not include these other two contributor variables in the regression.

[34] This is the correlation between the natural logarithm of the variables, since we use those in the analysis.

[35] Controlling for the project factors, the average degree of the contributors on a project and the average *betweenness centrality* of contributors on a project are negatively correlated with the success of a project.

7. Concluding Remarks

Knowledge spillovers are an important part of any learning or an R&D process. There are two possible mechanisms that facilitate such spillovers. One possibility is that an individual (or a firm) observes the outcome of an R&D effort of another individual, i.e., new technology or a patent, and learns about its own R&D process. A more direct mechanism is the interaction between different individuals who communicate with their colleagues, exchange emails, switch jobs and projects and collaborate in different research ventures. The first type of spillover is easier to model as a dynamic process in which any advance or success involving one project positively affects the success of related projects. The second type of learning spillover crucially depends on the specific network of interaction between individuals who are involved in the learning process. It is much more difficult to extract information regarding who talks with whom and how knowledge is shared between individuals. The OSS project network provides a unique opportunity for tracing such interactions and for examining the effect of the properties of the "collaboration" network on the success of different projects. A similar study can be done with respect to academic research in which it is possible to construct the network of collaboration. While the collaboration network has been constructed for different fields, it is important to take the next step and relate the properties of these collaboration networks to outcomes ("successes"), which can be measured, for example, by citations of different papers.

References

Ballester, A. Calvo-Armengol, A., and Y. Zenou (2006) "Who's Who on Networks: Wanted the Key Player" *Econometrica*, Vol.74 (5), 1403-1417.

D' Aspermont, C. and A. Jacquemin (1988), "Cooperative and Noncooperative R&D in Duopoly with Spillovers", American Economic Review, 78(5), 1133-1137.

Faust, K., and S. Wasserman, S., (1994), *Social Network Analysis: Methods and Applications*, Second Edition. New York and Cambridge, ENG: Cambridge University Press.

Freeman, L. (1979), "Centrality in Social Networks: Conceptual Clarification." *Social Networks*, 1: 215-239.

Goeree, J., McConnell, M., Mitchell, T., Tromp, T., and L. Yariv "Linking and Giving among Teenage Girls," mimeo 2007.

Goyal, S. (2007), *Connections: An Introduction to the Economics of Networks*, Princeton University Press.

Goyal, S., M. van der Leij, and J. Moraga, (2001), "R&D Networks," *Rand Journal of Economics* 32: 686-707

Goyal, S., and J. Moraga, (2006), "Economics: Emerging Small World" *Journal of Political Economy*, 114, 403-412.

Greene, W., (1993), "Econometric Analysis, Second Edition. New York: MacMillan Publishing Company.

Griliches, Z., (1986), "Economic Data Issues," in *Handbook of Econometrics*, Volume 3, Griliches, and M. Intriligator, editors. Amsterdam: North Holland Publishing Company.

Hann, I., Roberts, J., and S. Slaughter (2002) "Delayed Returns to Open Source Participation: An Empirical Analysis of the Apache HTTP Server Project, Carnegie Mellon University mimeo.

Harhoff, D., J. Henkel, and E. von Hippel (2003), "Profiting from voluntary spillovers: How users benefit by freely revealing their innovations, *Research Policy* 32: 1753-1769.

Hars, A., and S. Ou (2001), "Working for free? - Motivations for participating in open source projects," *International Journal of Electronic Commerce*, 6: 25-39

Hertel, G., Niedner, S. and S. Herrmann (2003), "Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux kernel," *Research Policy*, 32, 1159-1177.

Jackson, M.O., (2006), "The Economics of Social Networks," In *Proceeding of the 9th World Congress of the Econometric Society* (ed. R. Blundell, W. Newey and T. Persson). Cambridge University Press.

Jackson, M.O. (2008), "Social Networks in Economics", forthcoming in the *Handbook of Social Economics* (edited by Benhabib, Bisin and Jackson), Elsevier.

Lakhani, K., and R. Wolf (2005), “Why Hackers Do What They Do: Understanding Motivation and Efforst in Free Open Source Projects, In: Feller/Open, J. Fitzgerland, S. Hissam, K. Lakhani (eds.), Perspectives on Free and Open Source Software, MIT Press, Cambridge.

Lerner, J., and J. Tirole (2002), "Some Simple Economics of Open Source" *Journal of Industrial Economics*, 52: 197-234.

Raymond, E. (2000), “The Cathedral and the Bazaar", available at http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/.

Stallman, R., (1999), “The GNU Operating system and the Free Software Movement,” in Dibona, C., Ockman, S., and M. Stone editors, *Open Sources: Voices from the Open Source Movement,* O’Reilly, Sepastopol, California.

Appendix A: Tables

Table A1: Descriptive Statistics for 66,511 Projects with data all variables

VARIABLE	MEAN	STD. DEV.	MIN	MAX
Projects Not in the Giant Component (N= 47,814)				
downloads	10,959	938,658	0	2.00e+08
years_since	2.70	1.67	0	6.64
count_topics	1.51	0.81	1	7
count_aud	1.21	0.69	0	3
count_op_sy	2.08	1.58	1	21
count_trans	1.27	0.92	1	40
ds_1	0.25	0.43	0	1
ds_2	0.20	0.40	0	1
ds_3	0.20	0.40	0	1
ds_4	0.26	0.44	0	1
ds_5	0.21	0.41	0	1
ds_6	0.02	0.12	0	1
Inactive	0.02	0.14	0	1
Cpp	1.61	1.52	1	42
Degree	1.18	2.14	0	23
Star	0.08	0.28	0	1
Projects in the Giant Component (N= 18,697)				
Downloads	42,751	1,062,802	0	1.18e+08
years_since	3.63	1.70	0.08	6.65
count_topics	1.65	0.89	1	7
count_aud	1.34	0.70	0	3
count_op_sy	2.25	1.69	1	22
count_trans	1.38	1.66	1	45
ds_1	0.22	0.42	0	1
ds_2	0.17	0.38	0	1
ds_3	0.21	0.41	0	1
ds_4	0.30	0.46	0	1
ds_5	0.29	0.45	0	1
ds_6	0.03	0.17	0	1
Inactive	0.03	0.16	0	1
Cpp	3.84	6.72	1	338
Degree	6.26	8.53	1	299
Betweenness	0.00028	0.0015	0	0.12
Closeness	0.14	0.021	0.061	0.22
Star	0.45	0.49	0	1

Table A2(a): Correlation among all Variables: N=66,511

	ldown	lyears	lcpp	ldegree	ds1	ds2	ds3	ds4	ds5	ds6	inact	ltop	ltrans	los	laud
ldownloads	1.00														
lyears_since	0.29	1.00													
lcpp	0.23	0.18	1.00												
ldegree	0.24	0.22	0.44	1.00											
ds1	-0.38	0.03	0.00	-0.07	1.00										
ds2	-0.20	0.01	-0.01	-0.05	-.04	1.00									
ds3	0.04	0.03	-0.01	0.00	-0.2	-0.16	1.00								
ds4	0.25	0.04	0.03	0.05	-0.26	-0.24	-.19	1.00							
ds5	0.38	0.09	0.09	0.14	-0.23	-0.22	-.21	-.14	1.00						
ds6	0.11	0.06	0.04	0.07	-0.05	-0.05	-.05	-.05	0.01	1.00					
inactive	0.01	0.06	-0.01	0.02	-0.05	-0.04	-.05	-.06	-.05	-.01	1.00				
ltop	0.13	0.18	0.09	0.10	0.04	0.02	.04	0.06	0.08	.04	0.00	1.00			
ltrans	0.09	0.05	0.10	0.05	0.03	-0.01	-.03	0.05	0.08	.04	0.01	0.09	1.00		
los	0.12	0.29	0.09	0.05	0.04	0.02	0.01	0.02	0.05	.01	0.01	0.14	0.07	1.00	
laud	0.15	0.29	0.06	0.11	0.02	0.02	0.03	0.05	0.08	0.04	0.01	0.20	0.06	0.15	1.00

Note:

ltop = lcount_topics
ltrans= lcount_trans
los=lcount_op_sy
laud=lcount_aud

Table A2(b): Correlation among all centrality variables (Giant Component: N=18,697)

	lcpp	degree	lbetween	lcloseness	star
lcpp	1.00				
ldegree	0.49	1.00			
lbetween	0.71	0.64	1.00		
lcloseness	0.26	0.41	0.36	1.00	
star	0.17	0.74	0.26	0.27	1.00

Table A3: Regressions for projects at least two years old

Dept Variable: ldownloads	Regression '1' (All Projects)		Regression '2' (Giant Component)	
Independent Variables	Coeff.	T-stat	Coeff.	T-stat
Constant	-0.55	-5.98	4.60	7.22
lyears_since	2.21	36.06	2.27	19.97
lcount_topics	0.24	7.64	0.20	3.47
lcount_trans	0.38	10.15	0.39	6.34
lcount_aud	0.31	6.88	0.33	3.74
lcount_op_sy	0.15	6.90	0.21	5.32
ds_1	-2.12	-53.37	-2.08	-29.05
ds_2	-0.68	-16.07	-0.88	-11.26
ds_3	0.81	19.08	0.55	7.31
ds_4	1.84	45.84	1.68	24.31
ds_5	2.74	65.51	2.58	35.94
ds_6	2.14	23.12	2.00	13.76
inactive	0.45	5.31	0.41	2.68
lcpp	0.51	15.91	0.63	14.80
ldegree	0.21	8.05	-0.11	-2.24
giant_comp	-0.13	-2.06		
lgiant_cpp	0.40	8.88		
lgiant_degree	-0.067	-1.47		
betweenness			0.47	10.49
closeness			0.31	1.24
# of Observations	44,638		14,749	
Adjusted R-squared	0.40		0.38	

Table A4: Regressions for projects with more than one contributor

Dept Variable: Ldownloads	Regression '1' (All Projects)		Regression '2' (Giant Component)	
Independent Variables	Coeff.	T-stat	Coeff.	T-stat
Constant	0.51	5.53	4.04	7.99
lyears_since	1.60	36.71	1.74	23.60
lcount_topics	0.24	5.54	0.24	3.74
lcount_trans	0.41	9.12	0.44	6.80
lcount_aud	0.46	7.69	0.42	4.33
lcount_op_sy	0.13	4.28	0.16	3.47
ds_1	-2.13	-40.18	-2.17	-26.99
ds_2	-0.77	-13.65	-0.88	-10.12
ds_3	0.78	13.79	0.53	6.28
ds_4	1.89	35.76	1.75	22.56
ds_5	2.75	49.51	2.52	31.19
ds_6	2.05	16.46	1.83	11.17
inactive	0.28	2.12	0.36	1.80
Lcpp	0.46	7.94	0.75	13.90
ldegree	0.15	3.98	-0.14	-2.56
giant_comp	-0.60	-5.88		
lgiant_cpp	0.55	7.61		
Lgiant_degree	0.03	0.57		
betweenness			0.45	10.11
closeness			0.44	1.53
# of Observations	25,422		11,814	
Adjusted R-squared	0.43		0.40	

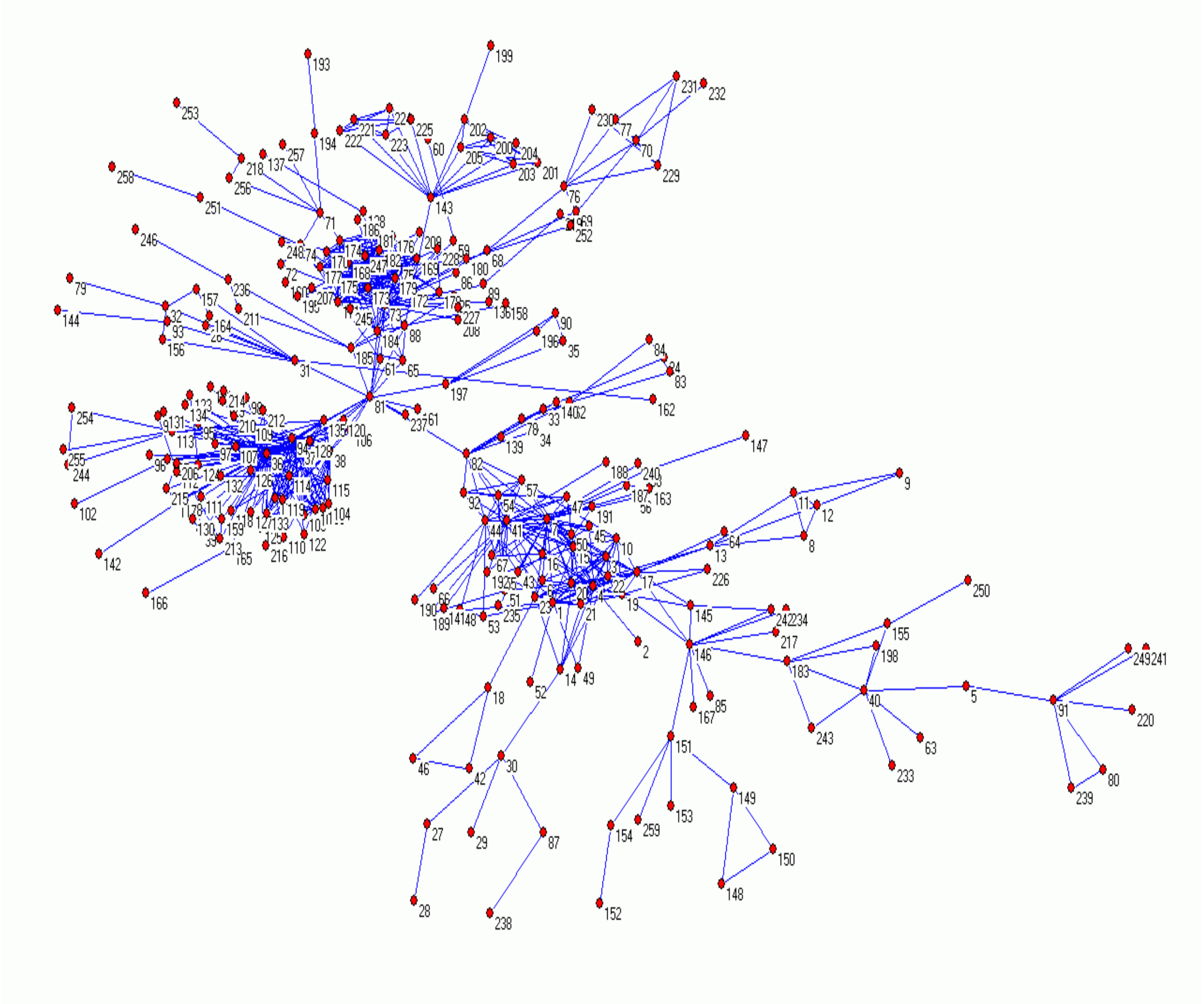

Figure 1: Projects in strongly connected component