THE PINHAS SAPIR CENTER FOR DEVELOPMENT
TEL AVIV UNIVERSITY

Microstructure of Collaboration: The Network of Open Source Software

Chaim Fershtman[1], Neil Gandal[2]

Discussion Paper No. 2-2008

March, 2008

*****We thank the Pinhas Sapir Center for financial support and Rafi Aviav for very helpful research assistance.

The paper can be downloaded from http://econ.tau.ac.il/sapir

[1] The Eitan Berglas School of Economics, Tel Aviv University, Tel Aviv 69978, Israel and CEPR, fersht@post.tau.ac.il.

[2] Department of Public Policy, Tel-Aviv University, Tel Aviv 69978, Israel and CEPR, gandal@post.tau.ac.il.

Abstract

Using data from source network at 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. We define a link between two projects if the projects have at least one contributor in common. Similarly two contributors are linked if they work together on at least one project. Interestingly, both the project network and the contributor network consist of one "giant" connected component and many smaller unconnected components. Knowledge spillovers may be closely related to the structure of such networks, since contributors who work several projects likely exchange information and knowledge. We thus examine the effect of the structure of these network on the success of OSS projects where success is defined as the number of downloads. Our main results are: (i) additional contributors are associated with an increase in output, but that additional contributors in projects in the giant component are associated with greater output gains than additional contributors in projects outside of the giant component; (ii) *Betweenness* centrality is highly associated with the number of downloads and this association is stronger than the relationship between other measures of centrality (*closeness* and *degree*) and the number of downloads. This result suggests 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.

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 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, while 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]

The research and development (R&D) process is affected by knowledge spillovers. Whenever co-workers collaborate on a joint project, they exchange information and knowledge. The phenomenon exists in commercial as well as in open source projects. Thus, the microstructure of the open source network might affect the R&D process and spillovers of knowledge. Studying the microstructure of the open source network can shed some light on the R&D process, and in particular on the R&D spillovers of knowledge 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.

firms and researchers.[5] (Add 4-5 references on papers on R&D with spillovers) When people interact, information is exchanged. 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.

There is a large economics literature that examines the properties of the networks and how they form, but this literature does not examine the relationship between network properties and output/success. Early studies included the Economics of "Network Effects," where the value of joining a network depends on the number of consumers that choose compatible products. This literature did not formally model the interaction among individual members of the network, since utility depended on just the total number of consumers on the network.[6] A more recent literature on social networks theoretically examines network formation where values differ for each link in the network and depend on how individuals are linked.[7] This framework enabled the examination of why (for example) buyer and seller may develop networks, rather than use a centralized market, and how such networks form.[8] A growing empirical literature examines key properties, such as the distribution of the degree of nodes in a network, and the average distance between pairs of nodes in social networks.[9] See Jackson (2005) for an overview of the literature on Social Networks in Economics.

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.[10] We use these data to examine how network structure affects output/success of open source projects. 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 and their roles in the project.[11] The Sourceforge.net information structure is rooted in projects. The data from SourceForge.net form a two-mode-network of projects and

[5] Goyal and Moraga (2001) examine the interaction between firm incentives to invest in R&D and the architecture of the collaboration network.
[6] See Farrell and Saloner (1985) and Katz and Shapiro (1986).
[7] See Jackson and Wolinsky (1996).
[8] See Kranton and Minehart (2001).
[9] See Jackson and Roberts (2007).
[10] These numbers are from June 2006 when we collected our data
[11] 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.

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. The links give rise to specific network structures. 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 network, the giant component contains 27,246 connected projects, while the second largest connected component consists of only 27 projects. Similarly, in the case of the contributor network, there is a giant component of 55,087 connected contributors and many smaller components. The second largest component in the contributor network consists of only 196 contributors.

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.[12] Hence, we assume that the number of downloads of open source projects is likely quite correlated with use and value.[13]

Our first important result is that additional contributors are associated with higher output, 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 on average for projects in the giant component.

We then examine how the network centrality measures such as *degree*, *betweenness* and *closeness* affect the number of downloads. Since these network centrality measures are only

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

[13] 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.

comparable across connected components, we conduct this analysis for projects in the giant component. We find that *Betweenness* centrality is highly associated with the number of downloads and this association is stronger than the relationship between other measures of centrality (*closeness* and *degree*) and the number of downloads. Since projects with higher values of *betweenness* are positioned in heavier information flows,[14] our results suggest that projects "well-positioned" in information flows are more successful. This result suggests that there are positive spillovers of knowledge for projects occupying critical junctures in the information flow.

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 (number of contributors, *betweenness*) 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, but that this increase is 150% greater for projects in the component with stronger ties, than other projects in the giant component.

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.[15] 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

[14] Airline hubs, for example, have relatively high values of betweenness.
[15] 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.

rooted in projects. The interface of SourceForge.net allows almost all of the information about the projects to be viewed by anyone.[16] 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 involves some investment of effort or time. A project is thus seen as a collaborative effort by its team members, or *contributors*.

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 1-mode networks; (i) 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.

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

- 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.[17] An additional 15% of the projects have two contributors. Hence, more than 80% of the projects have just one or 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" that work on many projects: 3,161 contributors worked on 5-9 projects, while 344 contributors worked on ten or more projects.

There are seven possible status of software development. 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.

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

Development status	Frequency in single contributor projects	Relative frequency	Frequency in multi contributor projects	Relative frequency
1 – Planning	11,687	21%	7,387	21%
2 - Pre-Alpha	9,121	17%	5,671	16%
3 – Alpha	9,767	18%	5,871	17%
4 – Beta	12,312	22%	8,014	23%
5 – Production/Stable	9,857	18%	7,252	20%
6 – Mature	806	1%	671	2%
Inactive	1,368	2%	654	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.[18] 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 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, similar to the contributor network, the project network also 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 payment or request permission in order to use them. One way to measure project success is to

[18] Hence, a contributor who has 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), the degree of that contributor would also equal four.

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. Downloads are also often used in order to measure the impact of academic papers and articles on the web.[19] 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 likely quite correlated with use and value.

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 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.[20] "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.[21] 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, of course, 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 perform robustness checks by conducting the analysis for projects that have been in existence for at least two years.

[19] Indeed, in a way academic papers are like open source projects. Typically, no permission or licensing agreement is required to access an academic publication.
[20] Given that there are only forty-two such "projects of the month," we cannot use this as our measure of success.
[21] The number of downloads is correlated with the stage of development.

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 that are defined for 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:

Control Variables:

- The variable *years_since* is the number of days 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. 25% of the projects were available written in a format that is independent of any operating system. *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.[22] *lcount_aud*=ln(*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.

Network Variables defined for all projects:

- The variable *cpp* is the number of contributors on 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 downloads and degree and the number of contributors depends of whether the project is inside or outside of the giant component. Hence, we also include the following interaction variables in the analysis:[23]

- giant_comp (1 if project in giant component, 0 otherwise)
- *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.

[22] It is typical to focus on the three main audiences. See Lerner and Tirole (2005). Nothing changes in main our results if we include niche audiences as well.

[23] The addition of different slopes for the control variables based on whether the project was inside or outside of the giant component had no effect on the main results regarding the number of contributors and the degree of the project. Hence, for ease of presentation, we do not include these variables in the regressions presented below in Table 5.

Network Variables that are comparable only among linked projects:

Betweenness: Being in the Center of the Information Flow

The "importance" of nodes in a network typically 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[24], 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.[25] *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*)[26]

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. The *closeness centrality*[27], or *closeness*, of a node is defined as the inverse of the sum of all distances between the node and all other nodes, standardized by multiplication with the number of other nodes, so that it lies in the range [0,1].[28] Formally, *closeness* is calculated as follows:

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

---

[24] Anthonisse (1971) and Freeman (1977) first quantified this notion.
[25] The denominator of (1) is the maximum possible value for the numerator, and thus standardizes the measure in the range [0, 1].
[26] The reason we add such a small number is because the mean value of *betweenness* is 0.00022.
[27] This definition, which is taken.
[28] See Faust and Wasserman (2005), p 184-185, and Sadibussi (1966).

C*loseness* measures how far each project is from the other projects in the network. We further define *lcloseness*=ln(0.05+*closeness*)[29]

We have data on all (114,406) observations for all of the network variables as well as on *years_since*.[30] The 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. Descriptive statistics of the variables are shown in Table A1 in the appendix.

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 $ldownloads_i = \alpha + \beta T_i + \gamma C_i + \delta Y_i + \varepsilon_i$, where the subscript *i* refers to the project. T_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.

As noted above, the data on the stage of development and the count variables are incomplete. Since there is no selection issue, one possibility would be to do what is typically done in such cases and ignore the 40% of the projects for which we do not have complete information.[31] Although we will do so at a later stage, we will first run a regression using the full data set (114,406 observations) with *ldownloads* as the dependent variable and the network variables and *years_since* as independent variables. Normally, running regressions without relevant independent variables is not a good idea, since it will lead to biased estimates of the coefficient of the included variables. This is not the case, however, if the included variables and the excluded variables are uncorrelated.

For the 66,511 observations for which there are data on all variables, the network variables (*ldgree* and *lcpp*) are not highly correlated with the excluded variables (the stage of development and count variables.) The highest pair-wise correlation (0.14) is between *ldegree* and ds_5. See Table A2 for correlations among all variables. Additionally, regressions of the network variables on the excluded control variables yield adjusted R-squared values of less than 0.05. Nevertheless, since the correlations are not exactly equal to zero, excluding the stage and count variables will introduce some bias in the estimated

[29] The reason we add such a small number is because the mean value of *closeness* is 0.14.
[30] There are 114,751 total projects, but we are missing data on downloads for a small number of them (345).
[31] See Griliches (1986) and Green (1993).

coefficients on the network variables. Hence, while we initially run regressions using all of the data, our preferred regressions are those that include only the observations with data on all relevant variables.. It is comforting to know that the main results are unaffected by whether we use the full data set, or the observations for which we have data on all relevant variables.

In section 4.1, we conduct an analysis on all 114,406 observations. In section, 4.2 we follow up this analysis be examining the giant component in detail, which enables us to include *betweenness* and *closeness*. Like section 4.1, we use data on all projects in the giant component. In section 4.3, we conduct an analysis similar to that of sections 4.1 and 4.2 using the subset of projects for which we have data on all variables.

4.1 Analysis Using All Projects

The results of a regression with all of the observations are shown in the first column of Table 5. 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 0.63. That is, a one percent increase in the number of contributors is associated with a 0.63 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: 1.18 (0.63+0.55). 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 (3.45 vs. 1.51). This means either (i) that contributors to projects in the giant component are inherently more skilled than the contributors who work on projects outside of the giant component or that (ii) there are knowledge spillovers among projects with ties that enhance the productivity of those who work together on these projects. It could also mean that some combination of these two effects is present.

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. The magnitude of the effect, however, is

different for the two groups. For projects outside of the giant component, the degree elasticity is 0.70, while the degree elasticity for projects in the giant component is 0.48. Both of these magnitudes are statistically significant from zero, and the difference in the magnitudes is statistically significant. The average degree for projects in the giant component (5.83) is much larger than the average degree for projects outside of the giant component (0.99). One possible explanation is that the marginal increase in the number of downloads associated with an increase in degree is lower for projects with a greater degree.

The estimated coefficient of *lyears_since* is positive (1.41) and statistically significant. This suggests that projects that have been active longer have more downloads, and the estimated coefficient (1.41) suggests that a doubling of the time a project has been active is associated with 141% more downloads. In section 4.4, we show that our main results regarding the association between the number of contributors and downloads and the association between the centrality measures (i.e., degree) and downloads is robust to excluding projects that are less than two years old.

4.2 Analysis for the Giant (Connected) Component

The first regression Table 5 suggests that there are differences between projects in the giant component and projects outside of the giant component. We now examine whether these differences can be explained (in part) by social network variables like *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 (0.90) is again statistically significant. The estimated contributor elasticity is, however, slightly lower than the estimated contributor elasticity for projects in the giant component using the full data set (1.18). The difference is likely due to the inclusion of *betweenness* and *closeness* in the regression.

The estimated *betweenness* elasticity (0.59) is positive and statistically significant. Thus, projects that sit in critical information flows have greater downloads. Similarly, the estimated *closeness* elasticity (1.14) is statistically significant as well: 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 matters for downloads, but how the projects are tied together and their position in the network. The estimated degree elasticity is smaller (0.10), although it is statistically significant as well. The smaller degree elasticity is

intuitive, since we are controlling for other measures of centrality (*betweenness* and *closeness*).

The regression has a relatively higher adjusted R-squared value (0.18) despite the fact that the explanatory variables in this regression include only network variables and the time for which the project has been active.

4.3 Analysis using all Variables – Complete Network

We continue the analysis for projects for which we have observations on all variables. In particular, when we include the stage and count variables, we are left with 66,511 projects. The estimates in the third regression in Table 5 are our preferred estimates for the whole network, since there is indeed correlation between the network variables (degree, the number of contributors) and the stage variables.

The results of the third regression in Table 5 show that our main result regarding the association between downloads and the number of contributors continues to hold. For projects outside of the giant component, the estimated "contributor" elasticity is 0.46 (versus 0.63 in the first regression in Table 5) and this effect is statistically significant. The estimated "contributor" elasticity is again virtually twice as large for projects in the giant coefficient: 0.90 (versus 1.18 in the first regression in Table 5). Further, the difference in the estimated "contributor" elasticity between projects in the giant component and projects outside of the giant component is statistically significant. The slight smaller numbers are likely obtained because the number of contributors is positively correlated with the stage of development.[32]

The estimated coefficients on degree for projects outside and inside the giant component are positive and statistically significant (0.19 and 0.14 respectively), but much smaller than in the first regression in Table 5 (0.70 and 0.48). The difference is again due to the fact that *ldegree* is positively correlated with the stage of development.

[32] The correlation between STAGE= ds_1 + 2*ds_2 + 3*ds_3 + 4*ds_4 + 5*ds_5 + 6*ds_6 and ldegree is 0.19. If y = a + bx + cz is the correct model (x,y,z are variables and a,b are parameters) and there is positive correlation between x and z and if b>0, the exclusion of z from the regression will lead to an upwardly biased estimate of b. Of course, in this case, there are more than two variables in the model. Hence, determining the direction of the bias is not that simple. Whether the count variables are included or not has virtually no effect on the estimates of degree and contributor elasticities.

The estimated coefficients on the stage variables have the expected 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.

4.4 Analysis using all Variables – Giant Component

We continue the analysis for projects for projects in the giant component for which we have observations on all variables. When we include the stage and count variables, we are left with 18,697 projects in the giant component. The estimates in the fourth regression in Table 5 are our preferred estimates for giant component, because of the positive correlation between the network variables (degree, the number of contributors) and the stage variables. When we conduct the analysis for the giant component only, the fourth regression in Table 5 shows that the estimated contributor elasticity is again statistically significant (0.61 versus 0.81 in the second regression in Table 5.)

The estimated *betweenness* elasticity is positive (0.49), statistically significant and not that much smaller than the estimate obtained in the second regression in Table 5 (0.59). While the estimated closeness elasticity is positive (0.38) and statistically significant at the 0.92 level, this estimate is much smaller than the estimate obtained in the second regression in Table 5 (1.14).[33] Although, these estimates are smaller than in the second regression Table 5, but they still suggests it is not just the ties among projects (via contributors) that matters for downloads, but how the projects are tied together and their position in the network: centrality matters.

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. This suggests that the two other centrality measures, especially *betweenness* are more important for the number of downloads that is the degree of the project.

[33] Again, the smaller estimates are likely because of the positive correlation between these centrality values and the stage of development.

Dept Variable: Ldownloads	Regression 1 All Projects		Regression 2 Giant Component		Regression 3 Projects with data on stage & count variables (Preferred Regression)		Regression 4 Giant Component Projects with data on stage & count variables (Preferred Regression)	
Independent Variables	Coeff.	T-stat	Coeff.	T-stat	Coeff.	T-stat	Coeff.	T-stat
Constant	0.60	23.42	7.68	14.67	0.72	17.76	5.71	10.76
lyears_since	1.41	72.97	2.04	43.75	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	0.63	27.55	0.81	22.38	0.46	18.71	0.61	16.71
ldegree	0.70	38.18	0.10	2.63	0.19	9.45	-0.13	-3.121
Giant_comp	-0.04	-0.87			-0.21	-3.86		
lgiant_cpp	0.55	15.88			0.44	12. 05		
Lgiant_degree	-0.22	-6.19			-0.05	-1.26		
betweenness			0.59	14.84			0.48	12.15
closeness			1.14	5.47			0.38	1.75
# of Observations	114,406		27,156		66,511		18,697	
Adjusted R-squared	0.13		0.18		0.41		0.41	

Table 5: Regression Results: Dependent Variable: ldownloads

4.5 Analysis for Established Projects

Nascent projects may not have reached a steady-state number of contributors. Personnel additions are probably more likely for relatively new products. It 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.[34] Our results are qualitatively unchanged.[35] In order to be concise, here we restrict our discussion to the preferred regressions – regressions 3 and 4 in Table 5.

In the case of the third regression, we are left with 45,188 observations (or 68% of the observations) when we restrict the analysis to projects that had been in existence for more

[34] The median age of projects in our data set as of June 2006 was 2.66 years.
[35] These results are available from the authors upon request.

than two years. For projects outside of the giant component, the estimated "contributor" elasticity is 0.50 (versus 0.46 in Regression 3 in Table 5), while the estimated "contributor" elasticity for projects in the giant coefficient is 0.90 (the same as in the third 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 third regression in Table 5. The estimated coefficients on the stage variables again have the expected signs.

When we run a regression analogous to the fourth regression in Table 5 for projects in the giant component that have been in existence for more than two years, we are left with 14,872 projects (or nearly 80% of the observations). The estimated contributor elasticity 0.64 (0.61 in the fourth regression in Table 5) is again positive, statistically significant and virtually unchanged. The estimated *betweenness* elasticity 0.46 (0.48) in the fourth regression in Table 5) is again positive, statistically significant and virtually unchanged. The estimated *closeness* elasticity (0.31) is positive, but smaller (0.38 in the fourth regression in Table 5), and is not statistically significant (t=1.21). The estimated *degree* elasticity is -0.10 (-0.13 in the fourth regression in Table 5) is virtually unchanged.

This analysis in this section reinforces our main results that suggest (i) that *betweenness* centrality is more highly associated with the number of downloads than other measures of centrality (*closeness* and *degree*) and that (ii) 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.

5. The Importance of Strong Ties

We defined two projects be linked if there was at least one contributor in common between them. In this section, we define two projects to be '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 that there is a link between two projects if there are at least two contributors who worked on both project.

Redefining the network has a dramatic effect on it structure. Previously there was a giant component of 27,246 projects when we defined projects as linked if they had at least on contributor in common. 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. No other strongly connected component has more than 27 projects. Figure 1 shows the projects in the most strongly connected component. A comparison of the median number of downloads between projects in the strongly connected component and other projects in the giant component suggests that a stronger connection results in more downloads. See Table 6.[36]

Group	# of projects	Mean # downloads	Median # downloads	95th percentile # downloads
Strongly Connected Component	259	82,238	2,035	374,419
Other Projects in Giant Comp.	26,897	30,230	98	35,924

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) the variable *l*strong_*degree* = *ldegree* strong*, and (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 thickly connected component and 1.56 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 fourth regression in Table 5.

[36] The same qualitative result obtains if we restrict the analysis to projects in stages 4-6. In this case, the projects in the thickly 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	5.63	10.60
lyears_since	1.67	31.18
lcount_topics	0.18	3.62
lcount_trans	0.43	7.92
lcount_aud	0.41	5.51
lcount_op_sy	0.18	4.94
ds_1	-2.02	-32.20
ds_2	-0.80	-11.89
ds_3	0.64	9.78
ds_4	1.78	29.10
ds_5	2.58	40.62
ds_6	2.02	15.33
inactive	0.35	2.51
lcpp	0.61	16.49
ldegree	-0.13	-3.07
thick_comp	-0.19	-0.23
lthick_cpp	0.97	3.00
lthick_degree	-0.56	-1.65
betweenness	0.47	11.92
closeness	0.37	1.73
# of Observations	18,687	
Adjusted R-squared	0.41	

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

References

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

Farrell, J. and Saloner, G. "Standardization, Compatibility and Innovation." RAND Journal of Economics, Vol. 16 (1985), pp. 70-83.

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

Goyal, S., and J. Moraga, 2001, "R&D Networks," Rand Journal of Economics 32: 686-707

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., "The Study of Social Networks in Economics," forthcoming in The Missing Links: Formation and Decay of Economic Networks, 2007, edited by Joel Podolny and James E. Rauch; Russell Sage Foundation.

Jackson, M., 2007, "Meeting Strangers and Friends of Friends: How Randon are Social Networks," *American Economic Review*, 97:890-915

Jackson, M. and A. Wolinsky, 1996, "A Strategic Model of Social and Economic Networks," *Journal of Economic Theory*, 71:44-74

Katz, M. and Shapiro, C., 1986, "Technology Adoption in the Presence of Network Externalities," *Journal of Political Economy*, 94, pp. 822-841.

Kranton, R., and D. Minehart, 2001, "A Theory of Buyer-Seller Networks," *American Economic Review*, 91:485-508

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.

Lerner, J., and J. Tirole (2005),

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, available at http://www.oreilly.com/catalog/opensources/book/stallman.html.

Appendix A: Tables

VARIABLE	MEAN	STD. DEV.	MIN	MAX
Projects Not in the Giant Component (N=87,250)				
Downloads	6368.93	694,983	0	2.00e08
years_since	2.54	1.61	0	6.81
cpp	1.51	1.49	1	152
degree	0.99	1.98	0	25
Projects in the Giant Component (N= 27,156)				
downloads	30,726	883,283	0	1.18e08
years_since	3.33	1.75	0.06	6.82
cpp	3.45	5.97	1	338
degree	5.83	7.84	1	299
betweenness	.000022	. 0013	0	.012
closeness	0.14	.021	0.061	0.22

Table A1a: Descriptive Statistics for all 114,406 projects

VARIABLE	MEAN	STD. DEV.	MIN	MAX
Projects Not in the Giant Component (N= 47,785)				
downloads	10,965	938,942	0	2.00e+08
years_since	2.76	1.71	0	6.81
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	0.13	0	1
inactive	0.02	0.14	0	1
cpp	1.61	1.52	1	42
degree	1.18	2.14	0	23
Projects in the Giant Component (N= 18,687)				
downloads	42,773	1,063,086	0	1.18e+08
years_since	3.63	1.74	0.08	6.82
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

Table A1b: Descriptive Statistics for 66,742 Projects with data all variables

	ldownloads	lyears_since	lcpp	ldegree
ldownloads	1.00			
lyears_since	0.26	1.00		
lcpp	0.25	0.16	1.00	
ldegree	0.26	0.20	0.42	1.00

Table A2a: Correlation among Variables: All 114,406 Observations

	ldownloads	lyears	lcpp	ldegree	lbetween	lcloseness
ldownloads	1.00					
lyears_since	0.31	1.00				
lcpp	0.32	0.16	1.00			
ldegree	0.23	0.17	0.46	1.00		
Lbetween	0.32	0.18	0.70	0.61	1.00	
Lcloseness	0.18	0.19	0.24	0.39	0.33	1.00

Table A2b: Correlation among Variables: Giant Component: 27,156 Observations

	ldown	lyears	lcpp	ldegree	ds1	ds2	ds3	ds4	ds5	ds6	inact	ltop	ltrans	lop	laud
Ldownloads	1.00														
lyears_since	0.29	1.00													
Lcpp	0.23	0.18	1.00												
Lodegree	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		
Lop	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

Table A2c: Correlation among all Variables: 66,742 Observations

Note:

inact=inactive
ltop = lcount_topics
ltrans= lcount_trans
lop=lcount_op_sy
laud=lcount_aud

	ldown	lyears	lcpp	ldegree	ds1	ds2	ds3	ds4	ds5	ds6	inact	ltop	ltrans	lop	laud
Ldownloads	1.00														
lyears_since	0.31	1.00													
Lcpp	0.30	0.16	1.00												
Lodegree	0.23	0.18	0.49	1.00											
ds1	-0.33	0.04	0.00	-0.05	1.00										
ds2	-0.21	0.02	-0.02	-0.04	0.03	1.00									
ds3	0.00	0.02	-0.02	-0.02	-0.15	-0.13	1.00								
ds4	0.23	0.06	0.04	0.05	-0.22	-0.22	-0.18								
ds5	0.38	0.12	0.13	0.14	-0.22	-0.21	-0.23	-0.16							
ds6	0.12	0.07	0.05	0.06	-0.04	-0.06	-0.06	-0.06	0.00	1.00					
Inactive	-0.02	0.04	-0.04	-0.02	-0.04	-0.03	-0.06	-0.08	-0.07	-0.02	1.00				
Ltop	0.13	0.19	0.11	0.07	0.07	0.03	0.05	0.08	0.08	0.04	0.00	1.00			
Ltrans	0.13	0.04	0.15	0.08	0.01	-0.02	-0.04	0.02	0.10	0.06	0.00	0.10	1.00		
Lop	0.13	023	0.11	0.04	0.05	0.04	0.04	0.04	0.06	0.04	-0.01	0.15	0.06	1.00	
Laud	0.16	0.30	008	0.10	0.05	0.04	0.01	0.06	0.08	0.06	-0.01	0.21	0.07	0.15	1.00
Lcloseness	0.18	0.21	0.26	0.41	-0.05	-0.03	-0.00	0.04	0.10	0.05	-0.03	0.07	0.04	0.07	0.04
Lbetween	0.31	0.18	0.71	0.64	-0.02	-0.02	-0.02	0.04	0.14	0.06	-0.02	0.10	0.13	0.09	0.07

	lbetween	lcloseness
lbetween	1.00	
lcloseness	0.36	1.00

Table A2d: Correlation among all Variables (Giant Component: 18,687 Observations)

Note:

inact=inactive
ltop = lcount_topics
ltrans= lcount_trans
lop=lcount_op_sy
laud=lcount_aud

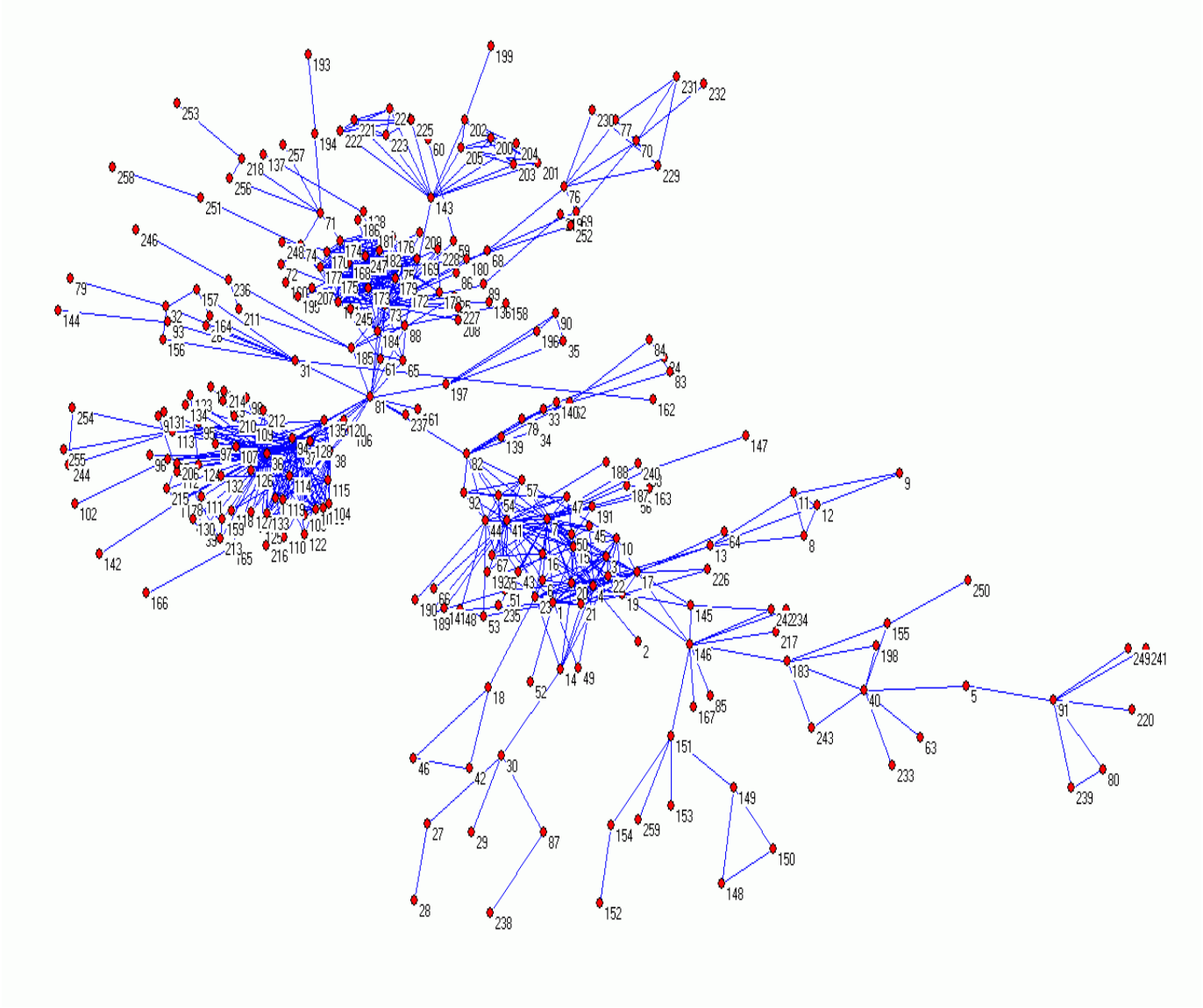

Figure 1: Projects in strongly connected component