MIS Quarterly

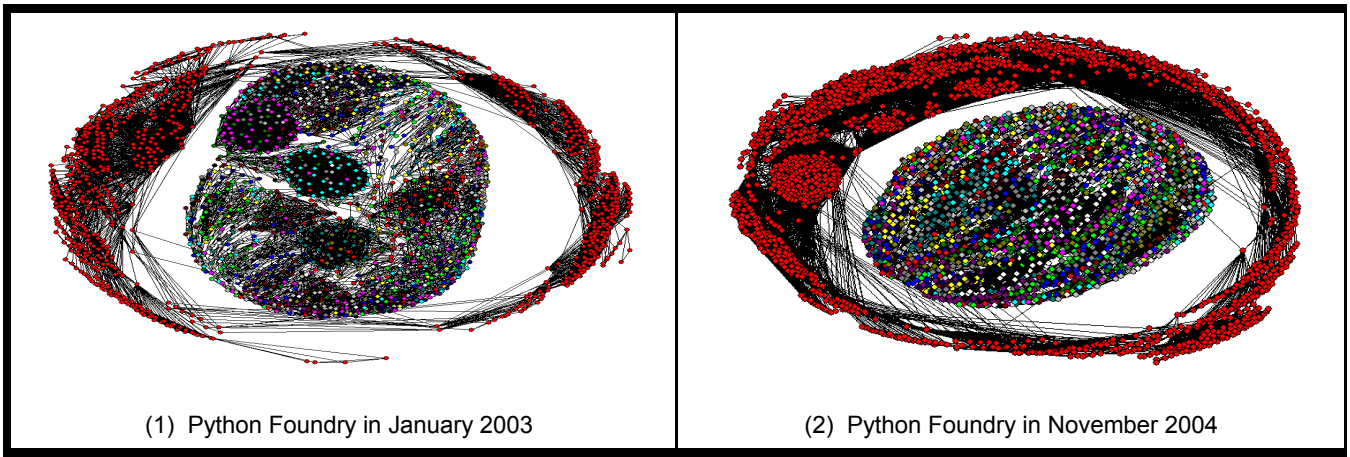
RESEARCH ARTICLE

NETWORK EFFECTS: THE INFLUENCE OF STRUCTURAL CAPITAL ON OPEN SOURCE PROJECT SUCCESS

Param Vir Singh
David A. Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213 U.S.A. {psidhu@cmu.edu)

Yong Tan
Michael G. Foster School of Business, University of Washington, Seattle, WA 98195 U.S.A. {ytan@u.washington.edu}

Vijay Mookerjee
School of Management, University of Texas at Dallas, Dallas, TX 52242 U.S.A. {vijaym@utdallas.edu}



(1) Python Foundry in January 2003

(2) Python Foundry in November 2004

Figure 2. Developer Collaboration Network Size and Component Structure (Developers are color coded by component. A color version of the figure is included with the online supplement. Isolates are not shown.)

Calculation of Indirect Ties

The frequency decay function is constructed on the argument that the rate at which the strength of the relation decreases with the increasing length of its corresponding path distance should vary with the social structure in which it occurs (Burt 1992). This decay function for developer i is given as

$$d_{ij} = 1 - f_{ij}/(N_i + 1)$$

where f_{ij} is the number of developers that i can reach within and including path length j, and N_i is the total number of developers that i can reach in the network. Then d_{ij} is the decay associated with the information that is received from developers at path length j. The frequency decayed indirect ties measure for developer i is then calculated as

$$\text{Indirect Ties FD}_i = \sum_{u=2}^{N} d_{ij} w_{ij}$$

where N is the total number of developers in the network and w_{ij} is the number of developers that lie at a path length of j from i. The larger the group over which a developer has to devote its time and energy, the weaker the relationship that it can sustain with any one member of the group, and the stronger the relationship with the closer ones. We divide this number by the number of project members to calculate a measure of frequency decayed indirect ties for a project.

Calculation of External Cohesion

Our measure of external cohesion for a developer is Burt's (1992) network constraint. It measures the extent to which a project member i's external network is invested in her relationship with external alter j. The constraint posed by external alter j on ego[1] i is measured as in Burt (2004) and averaged over all project members:

$$C_p = \sum_{i=1}^{N_P} \sum_{q=1}^{N_e} \left(p_{ij} + \sum_{q=1}^{N_e} p_{iq} p_{qj} \right)^2 \Big/ N_p, q \neq i, j$$

where N_p is the number of project members and N_e is the number of developers external to the project. There are two components to this constraint measure. First is the proportion of her total network time and energy that i directly allocates to external alter j

$$p_{ij} = (z_{ij} + z_{ji}) \Big/ \sum_{q=1}^{N} \left(z_{iq} + z_{qi} \right)$$

where z_{ij} is the tie strength between i and j. The second component is the strength of the indirect connections between i and j through mutual contacts q:

$$\sum_{q=1}^{N_e} p_{iq} p_{qi}$$

Here p_{iq} is the proportion of her total network time and energy that i devotes to q and p_{qj} is the proportion of her total network time and energy that contact q devotes to contact j. Note that contact q belongs to a group a developers that are external to the focal project. This formulation allows us to measure the extent to which a project member's external alters share relationships with each other. The higher values of constraint for a project imply that its external alters are more connected with each other. The higher the project's mean level of constraint, the greater its external cohesion and the lower the amount of global structural holes in its external network.

[1] In social network terminology, the focal actor is termed *ego* and the actors who have ties to the ego are termed *alters*.

Calculation of Technological Diversity

We first define the technological position of each project. Extant software engineering research suggests that the technological position of a software project can be defined on the following dimensions: type of the project (such as gaming or Internet applications, etc), programming language, user interface and operating system (Jones 1984; Sacks 1994). Each of these dimensions represents a different type of technical expertise. Project type represents the application domain knowledge whereas the other three represent the tools knowledge that comprises the knowledge of process, data, and functional architecture (Kim and Stohr 1998). Software engineering research has shown that the similarity of domain and tools affect the amount of knowledge that can be reused from one project to another (Banker and Kaufman 1991; Lee and Litecky 1999).

Following Jaffe (1986), we characterize a project's *technological position* by a vector $\mathbf{F}_p = (F_1 \ldots F_k)$, where k is the total number of categories under the four dimensions, and F_i is an indicator variable that equals 1 if the project p falls under category i. A project can fall under several categories within a single dimension. Technological diversity between the two projects p and q, is then calculated by the angular separation or uncentered correlation of the vectors $\boldsymbol{F}_p$ and $\boldsymbol{F}_q$ (Jaffe 1986):

$$\text{Technological Diversity}_{pq} = \mathbf{1} - \boldsymbol{F}_p \boldsymbol{F}_q' \Big/ \sqrt{\left(\boldsymbol{F}_p \boldsymbol{F}_p'\right)\left(\boldsymbol{F}_q \boldsymbol{F}_q'\right)}$$

Technological diversity varies from zero to one, with a value of one indicating the greatest possible technological diversity between two projects. This measure of diversity is purely directional; it is not affected by the length of the vector $\boldsymbol{F}$ and has been used in other studies (Jaffe 1986; Sampson 2007). We calculate the technological diversities of a focal project with all of the projects with which it shares a developer. We sum these measures and divide it by the number of such projects to calculate the technological diversity measure for the focal project.

Table A1 provides the descriptive statistics of the untransformed main variables. As a diagnostic test for the presence of multicolinearity, we calculated variance inflation factors (VIF) for each variable (Greene 2003). These factors measure how much of the variance of the estimated regression coefficients are inflated as compared to when the predictor variables are not linearly related. A maximum VIF value in excess of 10 is frequently taken as an indication that multicolinearity may be unduly influencing the estimates. The VIF of all predictor variables in the specified models were below 7, indicating that multicolinearity is not a problem.

Table A1. Descriptive Statistics of Key Independent Variables (*n* = 2,378; *Obs* = 3,850)

	Variable	Mean	Std	Min	Max	VIF
1	Repeat Ties	0.05	0.24	0.00	3.19	2.62
2	Repeat Ties Squared	0.61	0.50	0.00	10.18	1.92
3	Direct Ties	6.18	22.68	0.00	285.00	2.43
4	Indirect Ties FD	126.99	254.13	0.00	799.34	2.10
5	Direct × Indirect Ties FD	785.44	5764.59	0.00	227811.90	2.66
6	Project Size	2.86	5.66	1.00	198.00	1.90
7	External Cohesion	0.18	0.25	0.00	1.13	6.65
8	External Cohesion Squared	0.12	0.28	0.00	1.27	4.43
9	Technological Diversity	0.48	0.29	0.00	0.99	2.23
10	Technological Diversity Squared	0.25	0.17	0.00	0.79	2.20
11	Page Views	66589.28	763541.70	0.00	39004745.00	1.51
12	Bugs Closed	17.94	128.80	0.00	4831.00	1.29
13	Support	0.33	2.95	0.00	130.00	1.20
14	Pre sample CVS	409.65	3216.18	0.00	138928.00	1.79
15	Project Age	25.04	14.97	0.00	61.00	1.82
16	Project Age Squared	626.99	223.96	0.00	3721.00	1.23
17	CVS Commits	643.39	2623.55	0.00	41928.00	—

The mean, standard deviation, minimum and maximum values are provided for the untransformed variables where as the VIF are presented for the variables transformed as in Equation 1.

Table A2. Transformations Applied to Key Variables

No.	Variable Name	Transformation Applied
1	Repeat Ties	Mean Centered
2	Repeat Ties Squared	Square of 1
3	External Cohesion	Mean centered
4	External Cohesion Squared	Square of 3
5	Technological Diversity	Mean centered
6	Technological Diversity Squared	Square of 4
7	Direct Ties	Mean centered and scaled down by a factor of 100
8	Indirect Ties FD	Mean centered and scaled down by a factor of 1000
9	Direct X Indirect Ties FD	Product of transformed 9 and 10
10	Project Size	Log transformed
11	Pre sample CVS	Log transformed
12	Page Views	Log transformed
13	Support	Log transformed
14	Bugs Closed	Log transformed
15	Project Age	Mean centered and scaled down by a factor of 100
16	Project Age Squared	Square of 19
17	CVS Commits	Log transformed

Table A3. Correlation among Key Independent Variables (Variables are Transformed as in Equation 1) (*n* = 2,378; *Obs* = 3,850)

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***
1	1															
2	0.29	1														
3	0.27	0.17	1													
4	0.12	0.07	0.8	1												
5	0.19	0.11	0.43	0.17	1											
6	-0.19	-0.04	-0.38	-0.09	-0.65	1										
7	0.32	0.15	0.4	0.18	0.43	-0.31	1									
8	0.34	0.21	0.11	0.03	0.3	-0.32	0.3	1								
9	0.45	0.13	0.13	0.08	0.22	-0.18	0.33	0.34	1							
10	0.18	0.04	0	0	0.25	-0.25	0.16	0.09	0.1	1						
11	0.1	0.22	0.11	0.03	0.23	-0.24	0.22	0.15	0.13	0.31	1					
12	0.06	0.05	0.05	0.02	0.19	-0.2	0.11	0.09	0.06	0.41	0.31	1				
13	0.05	0.09	-0.01	0	0.08	-0.09	0.07	0.04	0.05	0.28	0.18	0.33	1			
14	-0.02	-0.06	0	0.01	-0.04	0.03	0.03	-0.01	-0.01	-0.01	0.33	-0.04	0	1		
15	0.02	0.06	0.06	0.01	0.2	-0.2	0.16	0.13	0.1	0.2	0.45	0.37	0.15	-0.09	1	
16	-0.03	-0.10	0.01	0.03	0.01	-0.01	0.12	0.04	0.04	0.06	0.15	0.03	0.04	0.07	0.34	1

The Numbers correspond to the variables names in Table A2

In the full sample high correlations among the network variables are due to a preponderence of zeros for isolate projects. On a subsample that included only those projects that have at least one project member working on an outside project, the correlations were significantly reduced as shown in Table A4.

Table A4. Correlation among Key Independent Variables (Variables are Transformed as in Equation 1) Only includes networked projects (*n* = 795; *Obs* = 1,257)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1															
2	0.16	1														
3	0.18	0.11	1													
4	0.09	0.08	0.65	1												
5	0.02	0.07	0.17	0.13	1											
6	0.02	0.01	-0.03	0.04	0.2	1										
7	0.25	0.13	0.24	0.13	0.27	0.01	1									
8	0.28	0.11	-0.12	-0.04	0.05	-0.06	0.15	1								
9	0.43	0.09	0.02	0.05	0.11	-0.01	0.29	0.32	1							
10	0.13	0.05	-0.27	-0.07	0.04	-0.03	0.02	-0.07	0.02	1						
11	0.06	0.12	-0.05	-0.02	0.11	-0.15	0.22	0.08	0.13	0.28	1					
12	0.02	0.07	-0.11	-0.01	0.08	-0.08	0.03	0	0.02	0.44	0.3	1				
13	0.03	0.07	-0.09	-0.02	0.03	-0.04	0.05	0.01	0.04	0.31	0.18	0.34	1			
14	-0.03	-0.02	0.05	0.03	-0.01	-0.01	0.1	0.01	-0.01	-0.01	0.22	0.01	0.02	1		
15	-0.04	-0.01	-0.08	-0.04	0.13	-0.15	0.16	0.11	0.1	0.19	0.5	0.35	0.15	-0.04	1	
16	-0.07	-0.09	-0.03	0.03	-0.03	-0.02	0.17	0.03	0.05	0.09	0.2	0.08	0.07	0.04	0.44	1

The Numbers correspond to the variables names in Table A2.

Hierarchical Bayes Estimation Procedure

$\theta_i = \{\beta_0, \beta_1, \ldots, \beta_{20}, \delta_1, \ldots, \delta_{ui-1}, \lambda_1, \ldots, \lambda_{ia-1}, \eta_1, \ldots, \eta_{ty-1}, \gamma_1, \ldots, \gamma_{os-1}, \tau_1, \ldots, \tau_{lt-1}\}$ represents the set of parameters that vary across projects (random effects parameters). Let $L(lnDV_{it})$ be the likelihood function for Equation 1 in the paper where i is project; t is network year. Further, we have $\theta_i = \boldsymbol{\nu}'\boldsymbol{Z}_i + \boldsymbol{\varepsilon}_{\theta i}$, where Z_i is a vector of ones, ν is the matrix of parameters which also represent the mean effect size, and $\varepsilon_{\theta i} \sim N(0, \Sigma_{\theta i})$.

The model is estimated using a standard MCMC hierarchical Bayes estimation procedure, using a Gibbs Sampler and the Metropolis Hastings algorithm coded in Matlab (Rossi et al 2005). In the hierarchical Bayes procedure, the first 100,000 observations were used as burn-in and the last 25,000 were used to calculate the conditional posterior distributions. The MCMC works as follows: MCMC recursively generates draw from the conditional distribution of the model's parameters.

$$\{\boldsymbol{\theta}_i\} \mid \boldsymbol{lnDV}_{it}, \boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{\vartheta}, \boldsymbol{\Sigma}_\theta$$
$$\boldsymbol{\vartheta} \mid \{\boldsymbol{\theta}_i\}, \boldsymbol{Z}, \boldsymbol{\Sigma}_\theta$$
$$\boldsymbol{\Sigma}_\theta \mid \{\boldsymbol{\theta}_i\}, \boldsymbol{Z}, \boldsymbol{\vartheta}$$

where $\boldsymbol{X}_i$ are the independent variables as defined in Equation 1.

Step 1

Generate $\{\boldsymbol{\theta}_i\}$.

$$f(\{\boldsymbol{\theta}_i\} \mid \boldsymbol{lnDV}_{it}, \boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{\vartheta}, \boldsymbol{\Sigma}_\theta) \propto N((\{\boldsymbol{\theta}_i\} \mid \boldsymbol{Z}_i, \boldsymbol{\vartheta}), \boldsymbol{\Sigma}_\theta) L(lnDV_i) \propto \mid \boldsymbol{\Sigma}_\theta \mid^{-1/2} exp(-1/2\ (\boldsymbol{\theta}_i - \boldsymbol{\vartheta}'\boldsymbol{Z}_i)' \boldsymbol{\Sigma}_\theta^{-1} (\boldsymbol{\theta}_i - \boldsymbol{\vartheta}'\boldsymbol{Z}_i)) L(\boldsymbol{\theta}_i - \boldsymbol{\vartheta}'\boldsymbol{Z}_i))$$

Metropolis-Hastings algorithm is used to draw from the conditional distribution of θ_i. To reduce the autocorrelation between draws of the Metropolis-Hastings algorithm and to improve the mixing of the MCMC we used an adaptive Metropolis adjusted Langevin algorithm (Atchade 2006).

Step 2

Generate ϑ

$$vec(\vartheta')|\{\theta_i\}, Z, \Sigma_\theta = MVN(u_n, V_m)$$

where $u_n = V_n\left(\left(Z' \otimes \Sigma_\theta^{-1}\right)vec(\Theta') + V_0^{-1}u_0\right)$,

$V_n = \left(\left(Z'Z \otimes \Sigma_\theta^{-1}\right) + V_0^{-1}\right)^{-1}$

u_0 and V_0 are prior hyper-parameters. We use diffuse prior for both these hyperparameters.
$Z = (Z'_1 \ldots Z'_N)$ is an N × nz matrix of covariates. nz is the dimension of Z_i
$\Theta = (\theta'_1 \ldots \theta'_N$ is an N × nθ matrix that stacks $\{\theta_i\}$. nθ is the dimension of θ_i matrix of covariates.
u_0 is set to $n\theta X1$ vector of zeros and $V_0 = 100\mathrm{I}_{n\theta}$

Step 3

Generate Σ_θ

$$\Sigma_\theta|\{\theta_i\}, Z, \vartheta \propto IW_{n\theta}\left(G_0^{-1} + \Sigma_{i=1}^{N}\left(\theta_i - \vartheta' Z_i\right)'\left(\theta_i - \vartheta' Z_i\right), f_0 + N\right)$$

where IW is inverse Wishart Distribution, and f_0 and G_0 are prior hyper-parameters. We use diffuse prior for both these hyperparameters. $f_0 = n\theta + 5$, and $G_0 = \mathbf{I}_{n\theta}$

Convergence Check: We follow the method suggested by Gelman and Rubin (1992) to check whether convergence has been achieved. The within to between variance for each parameter estimated across multiple chains was compared. Across five parallel chains, the scale reduction estimate for all parameters estimated was lower than 1.1, which indicated that the convergence was achieved.

Acceptance Rates for Metropolis Hastings: For Step 1, the acceptance rates achieved were approximately 19 percent.

References

Atchade, Y. 2006. "An Adaptive Version for the Metropolis Adjusted Langevin Algorithm with a Truncated Drift," *Methodology and Computing in Applied Probability* (8:2), pp. 235-254.

Banker, R., and Kauffman, R. 1991. "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," *MIS Quarterly* (15:3), pp. 375-401.

Burt, R. S. 1992. *Structural Holes*, Cambridge, MA: Harvard University Press.

Burt, R. S. 2004. "Structural Holes and Good Ideas," *American Journal of Sociology* (110:2), pp. 349-399.

Gelman, A., and Rubin, D. B. 1992. "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science* (7:4), pp. 457-472.

Greene, W. H. 2003. *Econometric Analysis* (6th ed.), Upper Saddle River, NJ: Prentice Hall.

Jaffe, A. B. 1986. "Technological Opportunity and Spillovers in R&D: Evidence from Firms' Patents, Profits and Market Value," *American Economic Review* (76), pp. 984-1001.

Jones, T. C. 1984. "Reusability in Programming: A Survey of the State-of-the Art," *IEEE Transactions on Software Engineering* (10:5), pp. 484-494.

Kim, Y., and Stohr, E. A. 1998. "Software Reuse: Survey and Research Directions," *Journal of Management Information Systems* (14:4), pp. 113-148.

Lee, N-Y, and Litecky, C. R. 1997. "An Empirical Study of Software Reuse with Special Attention to Ada," *IEEE Transactions on Software Engineering* (23:9), pp 537-549.

Rossi, P., Allenby, G., and McCulloch, R. 2005. *Bayesian Statistics and Marketing*, San Francisco: John Wiley & Sons.

Sacks, M. 1994. *On-the-Job Learning in the Software Industry*, Westport, CT: Quorum Books.

Sampson, R. C. 2007. "R&D Alliances and Firm Performance: The Impact of Technological Diversity and Alliance Organization on Innovation," *Academy of Management Journal* (50:2), pp. 364-386.