

THE IMPACT OF COMMUNICATION STRUCTURE ON ISSUE TRACKING EFFICIENCY AT A LARGE BUSINESS SOFTWARE VENDOR

Arne Beckhaus, SAP Research, arne.beckhaus@sap.com
Dirk Neumann, University of Freiburg, dirk.neumann@is.uni-freiburg.de
Lars M. Karg, SAP Research, lars.karg@sap.com

ABSTRACT

Software development's organizational design is often characterized by geographical dispersion and virtual team work. In this setup, communication is an essential task. We study this task by analyzing the association between communication structure and efficiency of the issue tracking process. This key process within software development is primarily carried out digitally and thus provides a suitable data set for analysis. We find that communication structure is associated with issue tracking efficiency in a case study at a large business software vendor.

Keywords: Communication, Issue Tracking, Efficiency, Social Network Analysis, Software Industry.

INTRODUCTION

The software industry is currently facing a transformation towards agile and lean software engineering principles [31]. Key elements of these concepts are close and informal collaboration and communication between team members [30]. Another observable trend in the industry is the increasing dispersion of work by means of globally distributed development and virtual team work. This evolution of the organization of work has been adopted from the experience of other industries [38]. As opposed to traditional organization forms, informal work gains importance in this setup [20]. This leads to challenges in determining the efficiency of work groups and processes. However, virtual work is often supported by technology which benefits data availability for research purposes. In the case of virtual work, communication data best reflects the informal collaboration patterns required to execute processes.

The scientific communities' interest in performance implications of group communication dates back to [25, 16]. These early works stemmed from the social sciences and conducted research in laboratory settings. They provided first evidence that communication structure does indeed have an influence on group performance. Later works from the field of social network analysis had a broader

scope but methodologically relied on surveying participants in order to build sociometric matrices that reflect communication and collaboration structure. For example, [3, 36] conduct studies in the field of academia. Here, student courses are studied by means of surveying network data and performance attributes from participants. Despite studying academic teams as well, [13] utilize the availability of electronic communication data which has recently become available due to the immense growth in usage of e-mail and online collaboration tools. With the availability of large data sources, researchers were able to enhance the debate by a social network analysis [39] perspective which is best applied on large networks [33]. Several works found network metrics in communication structure to be associated with group performance. Centrality [11] has been found to be a central concept of SNA. For example, [27] found that it is beneficial for coordination in the Enron e-mail dataset and [1] found that it is associated with the individual performance of members of virtual research and development groups. Other measures like network density have been found to be negatively associated with communication efficiency [40].

The availability of large electronic datasets did not only foster the use of social network analysis as research methodology, but also broadened the research scope. Many scholars followed von Hippel's and von Krogh's research call [18] to utilize the vast amount of data provided by the open source software development community for research in the field of organization science. This proposal is in line with [34, 35] who demand tighter links between organization science and information systems research. Research on the collaborative process of open source software development has largely concentrated on data provided by sourceforge.net, a leading hosting service which is perceived to be the most suitable data source for this kind of analysis [37, 40]. However, scholars have frequently addressed inter-project structure—e.g. which developers are active in which projects at the same time [28]. Intra-project communication has less frequently been analyzed (e.g. [7]) in both closed and open source development projects—especially with regard to process efficiency

instead of project success [26]. We therefore address the open research question whether intra-group communication structure is associated with the efficiency of the issue tracking process.

Network analysis on group level has become popular as opposed to the individual perspective taken by early works in the field of social network analysis [8]. This is in line with the observation that many organizations have re-organized their work around teams for reasons of flexibility and optimal utilization of resources [12, 23]. Software development is following this trend and is often carried out in teamwork [4, 17]. In order to maximize the concept's benefits, it is often exercised as virtual team work with geographically and culturally dispersed teams [38]. It is often the case, that this group work is not formally established but carried out in informal workgroups [20].

In this context of virtual group work, efficiency measures are hard to define. Managers need to abstract from evaluations of single employees and should rather look at processes. We therefore restrict our research question to communication in virtual work and analyze its association with efficiency of the software industry's issue tracking process.

The remainder of this paper is structured as follows. In the next section, we introduce our research hypotheses. Next, the issue tracking process is discussed and our research site is introduced. Finally, we present the results of our empirical analysis and close with a conclusion that also gives an outlook on future work.

RESEARCH HYPOTHESES

We propose three hypotheses in order to answer our research question whether communication structure and issue tracking efficiency are associated in the context of the software industry.

HYPOTHESIS 1: Communication Frequency is Positively Associated with Issue Tracking Efficiency

In the first hypothesis, we analyze the degree of communication between employees executing a process. This hypothesis is tailored to knowledge work which is the predominant type of work in the software industry. While an efficient process in the manufacturing industry is one with little manual interference and reliable automation, the software industry has not reached the same degree of process maturity yet [30]. Processes in the software industry must rather be seen as human centered and

collaborative. We assume that the degree of communication is positively correlated with process efficiency since communication is a central aspect of collaboration. While needing further refinement in a case study in order to adjust the metric to the application domain, we comprehend communication as exchange between various employees and not as intensity of dialogs. This hypothesis' suitability is backed by Bulkley & van Alstyne [5] who found that more frequent communication is positively associated with outcome.

HYPOTHESIS 2: Homogeneity in Communication Patterns is Positively Associated with Issue Tracking Efficiency

While Hypothesis 1 is intended to be analyzed on a single process execution level, our second hypothesis abstracts this perspective to a process participant level. Several executions of the same process are considered here. We believe this aggregation to be useful for two reasons. First, our case study based research must claim to have managerial implications. These are more expressive when aggregated over several process executions. Improvements are not able to be performed ex post on single process executions. They are rather applied to the process itself including all future runs. Therefore, aggregation of process execution data is useful for the derivation of more expressive improvements. Second, we believe that Hypothesis 2 is less prone to interfering effects than Hypothesis 1. There might be hidden causalities in the effect of communication on process efficiency of single process iterations that can be reduced by aggregation.

The metric analyzed on this aggregated level is the homogeneity of communication patterns. It says whether a process is executed in a more efficient way when all members and iterations communicate in a similar way. Despite again leaving the detailed definition of this metric to the context specific case study design, we advise to utilize a common statistical measure of homogeneity as the standard deviation and the context specific definition of communication used in Hypothesis 1. Gloor et al. [13] used a similar hypothesis and found that a balanced communication behavior correlates with performance.

HYPOTHESIS 3: Communication Network Structure is Associated with Issue Tracking Efficiency

Hypothesis 3 builds on hypothesis 2 as it also considers all employees active in the issue tracking process. Here, we analyze whether communication structure—measured by means of common metrics of

social network analysis—is associated with issue tracking efficiency. This hypothesis is formulated without an assumption regarding the sign of the correlation. Even more interesting than the question whether network structure boosts or hinders process efficiency is that whether a significant relationship exists at all. Mehra & Kilduff & Brass [29] have chosen a similar hypothesis and found that centrality as one common social network metric is associated with outcome.

In its combination, these three hypotheses provide a holistic perspective on the association between communication structure and issue tracking efficiency. All three hypotheses have been analyzed and confirmed in similar definitions in the related work.

ISSUE TRACKING

In the software industry, issue tracking is a central quality assurance process. When a tester experiences an unspecified behavior of a software product, he or she reports the incident in the so called issue tracking system. These systems are usually fully integrated solutions that comprise user management, issue report handling, search, and communication support [19].

At our research site, an issue report contains four data fields:

- The *software component*, where the tester believes that the problem might originate from,
- optionally the name of a *processor*, who should look into the issue next,
- a *priority* indicating the urgency of resolution according to defined service level agreements, and
- a *full text description* of the unspecified behavior containing information on how to reproduce the problem.

The issue tracking process is highly collaborative and usually involves different employees from reporting of the incident to resolution of the issue. An issue therefore consists of several actions that can be defined as elementary steps of collaboration. These actions are the main data source for our analysis since they reflect the communication between different employees that led to the resolution of an issue report.

According to [6], the issue tracking process consists of the following main steps:

- find a problem while using system,
- attempt to resolve problem,
- attempt to find workaround,
- diagnose the problem,
- design a fix for the bug,
- write the code for the fix, and
- recompile the module and link it with the rest of the system.

Figure 1 provides a fictitious example of an issue with six actions of three different employees. It demonstrates the existence of several informal roles present in issue tracking such as the reporter of the problem, a supportive employee that re-directs the issue, and a developer that finds a solution to the problem [22].

Most important for our analysis is the sequence of employees involved in the issue resolution. In an industrial setup with defined service level agreements specifying the maximal processing time (i.e. time from reporting to resolution of an issue), the process cannot solely rely on coincidental or voluntary participation. It rather allots groups of employees observing component queues. Since every issue needs to be assigned to a software component at all times, there is a responsibility for the corresponding group to process issues in their queue in a given amount of time. Therefore, it is not required to know a peer employee who is skilled in a certain matter, but to assign an issue to a software component only. However, the optional ‘processor’ field allows addressing a specific employee directly.

Based on this process definition, we are able to transform the technical issue tracking system into a suitable data base for our analysis. We restricted the corpus of 60 months of issue tracking data conducted in the analyzed application group to those issues that were resolved in order to be able to calculate the total processing time of the issue. This indicator is commonly used as a measure of the efficiency of issue processing (e.g. [9, 32, 2]).

In order to carry out communication network analysis, we need to define sub-groups in our application group. We chose the strategy to define employees who work on issues of a software component to belong to the same group. There is an overlap in this group definition which is unavoidable in the complex and networked software engineering process. However, overlapping groups are a well known and solvable problem in social network analysis (cf. [10]). In order to be able to perform this selection, we removed all issues from our data where the initial

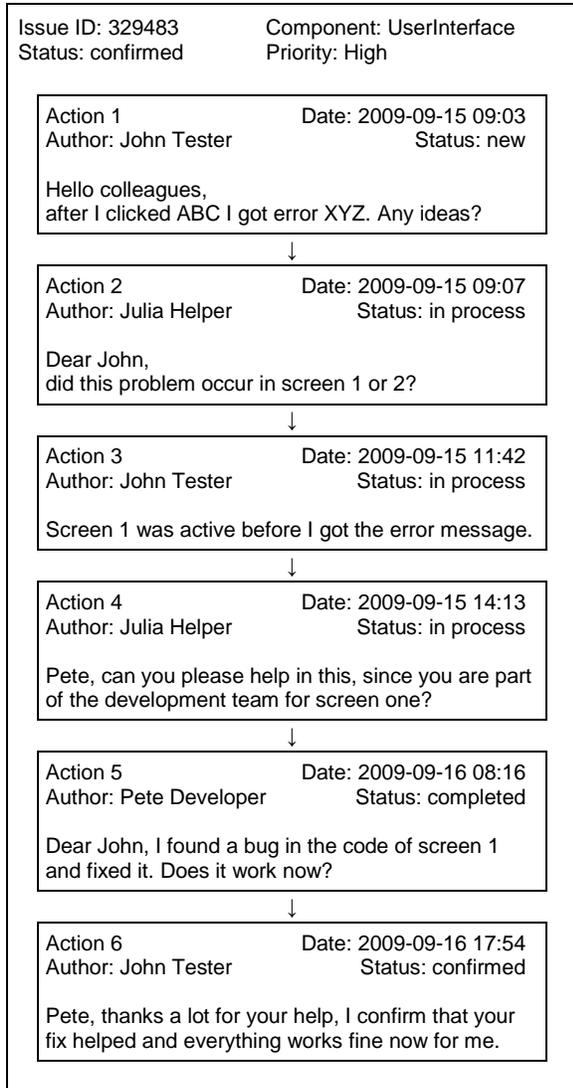


Figure 1. Sample Issue (fictitious data)

judgment of the affected software component was wrong and a different component was assigned in the course of processing. In order to omit very small groups, all group based analysis involved filtering according to a parametric minimal number of issues assigned to a software component.

RESEARCH SITE

In order to empirically validate our research hypotheses, we carried out a quantitative case study in the software industry. We cooperated with a leading European business software vendor.

Due to the size of the analyzed company that is a global player in the enterprise-resource-planning software market, it was first necessary to find a cooperating department that the case study could be

carried out in. Gathering data from all product lines and international affiliations seemed unreasonable. Instead, we focused our search on departments that are distinguishable from other groups within the company and have a sufficient size of 500 to 2000 employees. The selected group is a reasonable choice regarding these criteria, though we cannot provide identifying details due to confidentiality reasons.

ANALYSIS & RESULTS

All three research hypotheses use the same measure for issue tracking efficiency. We measure it as the issue's processing time as follows:

$$t(a_{compl}) - t(a_1),$$

where $a_i, i \in 1..n$ are the actions that the issue has taken and $a_{compl} = a_i$, where

$$i = \max(\{j \in \{1..n\} \mid status(a_j) = 'completed'\})$$

where t is a function that retrieves the date of an action and $status$ is a function that retrieves the processing status of an action. Most important about this definition is the fact that the corpus was restricted to confirmed issues, that is the problem was set to status 'completed' by a processor and to 'confirmed' by the original reporter. However, issues can be reopened (i.e. return to status 'in process' from 'completed' in the case of a negative confirmation by the reporter). Therefore, the processing time is defined to be the time between the reporting of the issue and the last solving action before being confirmed.

All three hypotheses are analyzed by means of correlation analysis. We calculated Pearson's correlation coefficient and validated the statistical significance (p-value) of our findings with the Pearson product-moment test [14].

HYPOTHESIS 1: Communication Frequency is Positively Associated with Issue Tracking Efficiency

In the context of the issue tracking process, the term communication can have different interpretations. Due to its collaborative nature, the entire process consists of communicative acts. An issue consists of several actions which can be either status changes or text messages. Each is an act of communication to colleagues with the common goal to resolve the problem. It is therefore important to find a context-

Table 2. Results of the Correlation Analysis of Hypothesis 1

Universal Set	$n = 26044$ issues
Pearson's Correlation Factor	$c = -0.24$
Statistical Significance	$p < 0.001$

specific, meaningful measure of the degree of communication frequency.

Trivially defining the communication frequency measure as the number of actions in an issue is not appropriate due to the causal impact on processing time which leads to a spurious correlation. Issues that require many steps to be solved process longer.

We defined a more suitable measure of communication in the issue tracking process. The communication index CI_i of an issue i is defined as the number of distinct processors of the issue divided by the number of actions of the issue. CI_i can take values between 0 (exclusive), representing a single soliloquizing processor, to 1 (inclusive), representing multiple processors each contributing to the issue exactly once.

In order to check the first hypothesis, we correlated all issues' communication index with their processing time. The results are shown in Table 2.

Processing time and communication index are negatively correlated. The correlation is statistically significant and sufficiently distinct in terms of an industrial data base that is seldom free of noise.

The negative sign of the correlation coefficient is expected and supports the hypothesis as formulated in this section's title, since higher communication frequency is reflected in high values of the communication index and higher efficiency in form of faster processing is reflected in low values of processing time. We therefore regard hypothesis 1 as being supported in our setup.

HYPOTHESIS 2: Homogeneity in Communication Patterns is Positively Associated with Issue Tracking Efficiency

In Hypothesis 2, we leave the issue based perspective and aggregate issues of identical software components to groups. As shown in the process description above, queues of issues assigned to software components are monitored by groups of employees. Therefore, we find that this grouping alternative is the one most closely aligned to our

Table 3. Results of the Correlation Analysis of Hypothesis 2

Universal Set	$n = 18$ groups
Pearson's Correlation Factor	$c = 0.56$
Statistical Significance	$p = 0.02$

process. This finding was confirmed by domain experts from the corresponding application group.

In a first attempt of group based correlation analysis, we tried to re-validate Hypothesis 1 by means of taking averages of the communication index and processing times of all issues belonging to a group. However, this variant of Hypothesis 1 is not correlated at a significance level of $p = 0.1$. One explanation is that taking the average of the communication index yields very similar values across different groups.

Despite the averaged communication index being similar across different groups, we observed that there is sufficient variation in the homogeneity of groups' communication patterns. Given a group G as a set of issues of the same software component, we define its communication homogeneity index as

$$HI_G = \sigma \left(\{ CI_i : i \in G \} \right)$$

where σ is the standard deviation function. Low values of HI_G represent homogeneously communicating groups whereas high values mean that different issues of a group are solved by means of different degrees of communication.

We checked Hypothesis 2 by correlating a group's communication homogeneity index with the average processing time of its issues. The universal set was derived by filtering those groups that contained less than 200 issues. The results are shown in Table 3.

Considering our industrial setup with noisy data, the second hypothesis' correlation can be interpreted to be strong. It is significant at the 5% level.

HYPOTHESIS 3: Communication Network Structure is Associated with Issue Tracking Efficiency

In our third hypothesis, we address more complex facets of communication by analyzing network structure. Related work has shown that the communicative ties between employees have an influence on performance. We validate these findings in our issue tracking database.

Table 4. Results of the Correlation Analysis of Hypothesis 3 with the social network metric betweenness centrality

Universal Set	$n = 188$ groups
Pearson's Correlation Factor	$c = -0.18$
Statistical Significance	$p = 0.01$

For this purpose, we adopt the issue grouping mechanism according to software components from Hypothesis 2. For each group G and the set of issues I_G therein, we aggregate the set E_G of employees participating in any of these issues and the sets $E_{G,i}$ of employees participating in issue I of group G . We then build the social network graph SN_G of a group as

$$SN_G = \begin{cases} v_{1,1} & \cdots & v_{1,|E_G|} \\ \vdots & \ddots & \vdots \\ v_{|E_G|,1} & \cdots & v_{|E_G|,|E_G|} \end{cases}$$

where

$$v_{i,j} = |h_i^j|: i \neq j, h \in I_G, i \in E_{G,h}, j \in E_{G,h} [$$

In short, a relation between two employees is valued with the number of issues they have jointly worked on.

Building on this graph definition, we have applied common measures in the field of Social Network Analysis on the graph of each group and correlated their outcome with the group's issues' average processing time. The universal set was derived by filtering those groups that contained less than 20 issues. The filter was chosen to allow for smaller groups compared to Hypothesis 2 since network metrics can already be calculated on groups of this size.

The results with the choice of the social network metric betweenness centrality are shown in Table 4. The choice of a centrality metric is frequent in related work. For example, Sparrowe et al. [36] state: "Centrality, the extent to which a given individual is connected to others in a network, is the structural property most often associated with instrumental outcomes."

The results with the choice of the social network metric Krackhardt efficiency [24] are shown in Table 5. It is specifically designed for informal organization structures and therefore matches our process orientation paradigm. A social network is efficient in Krackhardt's definition when there are just as many

Table 5. Results of the Correlation Analysis of Hypothesis 3 with the social network metric Krackhardt efficiency

Universal Set	$n = 188$ groups
Pearson's Correlation Factor	$c = -0.31$
Statistical Significance	$p < 0.001$

connections as required to weakly connect all members.

Both metrics show a significant correlation. Without going into a detailed analysis of the metric's interpretation which would be beyond the scope of the hypothesis' formulation, the analysis supports our hypothesis that communication network structure is associated with issue tracking efficiency.

CONCLUSIONS

In this paper, we analyzed the effect of communication structure on issue tracking efficiency in the software industry. We empirically evaluated our research question in a case study with a leading business software vendor.

We found that communication frequency is associated with issue tracking efficiency. When viewing multiple process executions, those groups of issues are most efficiently resolved that have a homogenous communication pattern. The analysis of communication network structure revealed that betweenness centrality and Krackhardt efficiency are positively associated with issue tracking efficiency.

We identified some research challenges that are partially threads to the validity of our work. On the one hand, our data might not cover all communication of the issue tracking process. We cannot grasp e-mail, telephone, or face to face communication and might therefore have a bias in our data (cf. [21, 15]). However, domain experts say that the fraction of missed communication is low, basically due to the fact that this offline communication is often supplemented in the system for documentation reasons.

This leads to the major direction of our future work which is the advancement of the approaches to derive communication networks implicitly coded in existing corporate data bases. We will experiment with different network definitions and also utilize more concepts of social network analysis. Additionally, we will develop more sophisticated ways to bridge the

group based network analysis techniques and our process oriented research question.

REFERENCES

1. Ahuja, M. K., Galletta, D. F., and Carley, K. M. (2003). Individual Centrality and Performance in Virtual R&D Groups: An Empirical Study. *Management Science*, 49 (1), 21–38.
2. Au, Y. A., Carpenter, D., Chen, X., and Clark, J. G. (2009). Virtual organizational learning in open source software development projects. *Information & Management*, 46 (1), 9–15.
3. Baldwin, T. T., Bedell, M. D., and Johnson, J. L. (1997). The social fabric of a team-based M.B.A. program: Network effects on student satisfaction and performance. *Academy of Management Journal*, 40 (6), 1369–1397.
4. Brodbeck, F. (2001). Communication and performance in software development projects. *European Journal of Work and Organizational Psychology*, 10 (1), 73–94.
5. Bulkley, N. and van Alstyne, M. W. (2006). An Empirical Analysis of Strategies and Efficiencies in Social Networks. Technical report, MIT Sloan Research Paper No. 4682-08.
6. Crowston, K. (1997). A Coordination Theory Approach to Organizational Process Design. *Organization Science*, 8 (2), 157–175.
7. Crowston, K., Wei, K., Li, Q., and Howison, J. (2006). Core and Periphery in Free/Libre and Open Source Software Team Communications. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, IEEE Computer Society, Washington, DC, USA.
8. Cummings, J. N. and Cross, R. (2003). Structural properties of work groups and their consequences for performance. *Social Networks*, 25 (3), 197–210.
9. Das, A. (2003). Knowledge and Productivity in Technical Support Work. *Management Science*, 49 (4), 416–431.
10. Davis, G. B. and Carley, K. M. (2008). Clearing the FOG: Fuzzy, overlapping groups for social networks. *Social Networks*, 30 (3), 201–212.
11. Freeman, L. C. (1979). Centrality in Social Networks: Conceptual clarification. *Social Networks*, 1 (3), 215–239.
12. Glassop, L. (2002). The organizational benefits of teams. *Human Relations*, 55 (2), 225–249.
13. Gloor, P. A., Paasivaara, M., Schoder, D., and Willems, P. (2008). Finding Collaborative Innovation Networks Through Correlating Performance With Social Network Structure. *International Journal of Production Research*, 46 (5), 1357–1371.
14. Greene, W. H. (2007). *Econometric Analysis*. Prentice Hall, 6th edition.
15. Grippa, F., Zilli, A., Laubacher, R., and Gloor, P. A. (2006). E-mail May Not Reflect The Social Network. In *NAACSOS Conference*, 22–23 June, North American Association for Computational Social and Organizational Science, Notre Dame, IN.
16. Guetzkow, H. and Simon, H. A. (1955). The impact of certain communication nets upon organization and performance in task-oriented groups. *Management Science*, 1 (3/4), 233–250.
17. He, J., Butler, B. S., and King, W. R. (2007). Team Cognition: Development and Evolution in Software Project Teams. *Journal of Management Information Systems*, 24 (2), 261–292.
18. von Hippel, E. and von Krogh, G. (2003). Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*, 14 (2), 209–223.
19. Johnson, J. and Dubois, P. (2003). Issue tracking. *Computing in Science & Engineering*, 5 (6), 71–77.
20. Kane, G. (2007). IS Proficiency in Social Networks. In *ICIS 2007 Proceedings*.
21. Kerr, D. S. and Murthy, U. S. (2009). The effectiveness of synchronous computer-mediated communication for solving hidden-profile problems: Further empirical evidence. *Information & Management*, 46 (2), 83–89.
22. Koru, A. and Tian, J. (2004). Defect handling in medium and large open source projects. *IEEE Software*, 21 (4), 54–61.
23. Kozlowski, S. W. and Ilgen, D. R. (2006). Enhancing the Effectiveness of Work Groups and Teams. *Psychological Science in the Public Interest*, 7 (3), 77–124.
24. Krackhardt, D. (1994). Graph Theoretical Dimensions of Informal Organizations. In *Computational Organizational Theory* (Carley, K. M. and Prietula, M. J., eds.), 88–111, Lawrence Erlbaum Associates, Hillsdale, NJ.
25. Leavitt, H. J. (1951). Some effects of certain communication patterns on group performance. *Journal of Abnormal and Social Psychology*, 46 (1), 38–50.
26. Lee, S.-Y. T., Kim, H.-W., and Gupta, S. (2009). Measuring open source software success. *Omega*, 37 (2), 426–438.
27. Liaquat, H., Wu, A., and Choi, B. (2006). Measuring Coordination through Social Networks. In *ICIS 2006 Proceedings*.
28. Madey, G., Freeh, V., and Tynan, R. (2004). Modeling the Free/Open Source software

- community: A quantitative investigation. In *Free/Open Source Software Development* (Koch, S., ed.), 203–221, Idea Group Publishing, Hershey, Pennsylvania.
29. Mehra, A., Kilduff, M., and Brass, D. J. (2001). The Social Networks of High and Low Selfmonitors: Implications for Workplace Performance. *Administrative Science Quarterly*, 46, 121–146.
 30. Messerschmitt, D. G. and Szyperski, C. (2005). *Software Ecosystem: Understanding an Indispensable Technology and Industry*. MIT Press.
 31. Middleton, P. and Sutton, J. (2005). *Lean Software Strategies: Proven Techniques for Managers and Developers*. Productivity Press.
 32. Midha, V. (2008). Does Complexity Matter? The Impact of Change in Structural Complexity on Software Maintenance and New Developers' Contributions in Open Source Software. In *ICIS 2008 Proceedings*.
 33. Newman, M., Barabási, A. L., and Watts, D. J. (2006). *The Structure and Dynamics of Networks*. Princeton University Press.
 34. Orlikowski, W. J. and Barley, S. R. (2001). Technology and Institutions: What Can Research on Information Technology and Research on Organizations Learn from Each Other? *MIS Quarterly*, 25 (2), 145–165.
 35. Picot, A. and Baumann, O. (2009). The Relevance of Organization Theory to the Field of Business and Information Systems Engineering. *Business & Information Systems Engineering*, 1 (1), 62–69.
 36. Sparrowe, R., Liden, R., Wayne, S., and Kraimer, M. (2001). Social Networks and the Performance of Individuals and Groups. *Academy of Management Journal*, 44 (2), 316–325.
 37. Tan, Y., Mookerjee, V., and Singh, P. (2007). Social Capital, Structural Holes and Team Composition: Collaborative Networks of the Open Source Software Community. In *ICIS 2007 Proceedings*.
 38. Wakefield, R. L., Leidner, D. E., and Garrison, G. (2008). A Model of Conflict, Leadership, and Performance in Virtual Teams. *Information Systems Research*, 19 (4), 434–455.
 39. Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.
 40. Wu, J. and Goh, K. Y. (2009). Evaluating Longitudinal Success of Open Source Software Projects: A Social Network Perspective. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*.