

The Communication Patterns of Technical Leaders: Impact on Product Development Team Performance

Kate Ehrlich
IBM Research
Cambridge, MA 02142
katee@us.ibm.com

Marcelo Cataldo
ConnReperio, LLC
New York, NY 11101
mcataldo@connreperio.com

ABSTRACT

Leaders are important for the smooth functioning of teams but their effect on team performance is less well established. Based on a sample of 55 software development teams from two companies we examined the association between technical leaders' communication structure and the team's performance. Consistent with past research, we found that a team's productivity and quality improve when the team communication follows hierarchical and small-world patterns, respectively. However, there was significant additional improvement in both productivity and quality when technical leaders shared more information than they gathered, and when they were more central in the communication network. Surprisingly they were not necessarily the most "central" individuals in the team's communication structure. We discuss the implication of these results for new tools that provide feedback to leaders and the team.

Author Keywords

Communication; Technical Leadership; Hierarchy; Centrality; Social Network Analysis; Teams

ACM Classification Keywords

H.5.3 [Group and Organization Interfaces]: Asynchronous Communication, Computer-supported Cooperative Work;

General Terms

Human Factors, Management, Measurement, Performance.

INTRODUCTION

Product development teams face the challenge of delivering high quality work on time and within budget while navigating uncertainty, conflicting goals and distributed work (e.g. [2]). It is not surprising that products are frequently delivered late, over budget and with poor quality [35]. It is generally acknowledged that the success of product development is dependent on informal communication for building the shared knowledge that is necessary for coordination especially in distributed teams [13, 23]. However, responsi-

bility for keeping teams on track falls to technical team leaders who define and design the core technical elements and direct the team. These leaders help members identify and leverage expertise that is distributed across the team members to apply the knowledge to the tasks at hand [19]. They must coordinate separate, but inter-dependent, tasks [9] while also allowing individuals to pursue their own work with minimal distractions (e.g. [10]). They also must support collaboration despite differing opinions, backgrounds and perspectives amongst team members [6, 17].

Yet despite their important role, there has been scant research on the effect of technical leaders on team performance. Research in other settings suggests that leadership can influence the flow of information and consequently impact project performance [1, 18, 37]. In product development teams, technical leaders have an important role in coordination of work within and across teams [9]. However, that research did not look at the implications for team performance. Investigations of sales and other types of teams suggest that teams perform better when leaders are central within a social network [3, 28]. However, this research has not shed light on how leaders are positioned within the flow of communication and how their position may be associated with team performance.

This paper reports the results of a study that examined the communication structure of technical leaders, and whether their position in the flow of information is associated with delivery and quality product performance. We performed our analyses using archival data from 55 teams working on large-scale development projects from two companies.

PREVIOUS RESEARCH

Product Development Tasks

Product development is highly complex and the impact of leadership styles and communication patterns is dependent on the kind of task being performed. Cataldo and Ehrlich [8] distinguished between tasks such as iteration planning which contributes to the **delivery performance** of a project, and core development which contributes to the number of defects and hence the **quality** of the software.

Delivery performance involves planning and executing development activities. Communication within the team needs to be sufficient to keep team members informed and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CSCW'14, February 15–19, 2014, Baltimore, Maryland, USA.
Copyright © 2014 ACM 978-1-4503-2540-0/14/02...\$15.00.

aligned with the current set of goals without distracting them from the tasks. Cataldo and Ehrlich [8] argued, and found, that a team benefitted when there was a hierarchical communication structure in which information flows out from a core person without flowing back (see also [26])

Quality performance, on the other hand, emphasizes the identification and integration of the right knowledge to make wise design or implementation decisions that minimize defects [10, 16]. Project members might struggle to make sense of new knowledge and effectively integrate it into the relevant context because of differences in perspectives, lack of common understandings and lack of familiarity working together [6]. Failure to adequately make sense of the knowledge and information can result, for instance, in unidentified dependencies, which lead to coordination breakdowns and ultimately, lower product quality [10]. Cataldo and Ehrlich [8] found that a hierarchical communication structure, which benefitted delivery performance, was detrimental to quality performance. Rather, quality outcomes benefitted from a small-world communication structure, which is characterized by small dense clusters of people who communicate frequently and few connections among those clusters. This communication structure brings about higher shared knowledge and greater trust amongst team members which makes it easier to integrate knowledge and access information more rapidly [7, 36].

The present study builds on this research to examine whether the position of the technical leader within the flow of communication has any additional contribution to the performance of the team over and above the effect of the team's communication structure. Specifically, we examine the effect on the team's delivery performance and quality performance, of technical leaders who are (a) centrally positioned, overall, within the team's communication network, (b) positioned to receive information from team members (focus on in-flow communication), and (c) positioned to share information with team members (focused on out-flow communication).

Leadership and Network Centrality

Research on the social network of team leaders in organizations found that leaders' centrality is associated with better team performance [3, 4, 28]. For instance, Mehra and colleagues [28] found that sales teams, where leaders were located within the center of the network, had higher sales and higher customer loyalty.

It is not clear whether we should expect similar results for product development teams, which typically have a much flatter organizational structure than other teams. Moreover, product team members are engaged in tasks with a high degree of inter-dependency that are generally resolved at the individual rather than the team level [10] and therefore may not benefit from any additional direction that comes from the technical leaders. According to this argument, technical leaders may not be more central in the network of successful teams than other team members.

On the other hand, technical leaders are crucial for coordination both internally and externally in product development projects, especially large-scale projects (e.g. [1, 9]). Technical leaders who are centrally located within a communication network are better positioned to collect and disperse the information necessary for coordination [24] and therefore in a better position to help mobilize the resources in the team.

RQ1a: Will the centrality of technical leaders be positively associated with team performance in delivery outcomes?

RQ1b: Will the centrality of technical leaders be positively associated with team performance in quality outcomes?

Leaders as Receivers of Information

There are further insights to be gained by studying the directionality associated with network position. In a communication network, leaders can be positioned as receivers of information (in-flow) when team members send information to them. Alternatively, they can be positioned as providers of information to the team (out-flow).

Technical leaders may be centrally positioned within the in-flow of communication because of their technical reputation. Research suggests that teams perform better when the leader is perceived as having a stronger and more positive reputation [28, 40]. Additionally, technical leaders who receive information from more team members are better informed about the on-going work and better positioned to integrate diverse information. This argument suggests that teams will benefit when the technical leaders are positioned within the incoming flow of information.

On the other hand, product development is an information intensive activity involving a high degree of interdependence between tasks. Developers may resolve these dependencies locally without the need for additional intervention by technical leaders [10]. This argument suggests that the team's performance is not associated with the position of the leader in the in-flow of information.

RQ2a: Will in-degree centrality of technical leaders be positively associated with team performance in delivery outcomes?

RQ2b: Will in-degree centrality of technical leaders be positively associated with team performance in quality outcomes?

Leaders as Providers of Information

A different model of technical leadership borrows from research on leader-member exchange theory (LMX). Recent research on the antecedents of LMX demonstrated that relational behaviors such as supporting, recognizing, consulting and delegating, produce higher quality leader-member relationships, and, by extension, higher group performance [40]. Similarly, others have found evidence that leaders have better team outcomes when they sustain strong

personal relationships by integrating and combining contributions from team members [39].

Research with software development teams find that these teams perform better with empowering leaders who encourage self-development, better teamwork and who participate in individual goal setting [18]. Further, performance is improved when leaders are able to share information that is relevant to the context of each team and their needs [37]. This result is consistent with a perspective in which product development teams benefit from having leaders who are centrally positioned within the out-flow of information.

On the other hand, an overly directive leadership may stifle an individual's ability to coordinate directly with others.

RQ3a: Will out-degree centrality of technical leaders be positively associated with team performance in delivery outcomes?

RQ3b: Will out-degree centrality of technical leaders be positively associated with team performance in quality outcomes?

RESEARCH SETTING

We examined the proposed hypotheses using data collected from two product development endeavors from two different companies. We collected data from the development of the first release of the Jazz software development environment on which the IBM^R Rational^R Team ConcertTM (RTC) products are based¹. We will refer to this dataset as Project A. The second dataset corresponds to the development effort associated with the 9th release of a complex embedded system produced by a company operating in the automotive industry. We will refer to this dataset as Project B. The following paragraphs describe both projects in more detail.

Project A

Our analysis of the RTC project focused on the 13 months prior to the release of the product's first version, which was the period of time between July 2007 and August of 2008. The time period was chosen because prior to July 2007 the amount of development activity was minimal and on that month the project underwent a significant change in process opening up the development process to the Jazz community. A total of 161 developers, worked on the project during the period analyzed. There were 26 functional teams distributed in 21 different sites across the U.S., Canada and Europe. Teams were responsible for one or more components of the system. Project members belonged to one or more of these teams. In addition, individuals have roles within each team and those roles were made transparent to all project members via the role definition capabilities of the RTC tool. The defined roles were "component lead", "contributor", "part-time contributor", "project admin",

"connector admin", "dashboard admin", "integration stream admin" and 10 more roles, which were specialized variants of the "contributor" role.

The project organized its development effort using iterations, which are short-lived (e.g. 6 to 8 weeks) units of development activities. The tasks to be performed in each iteration and by each development team were planned at the beginning of each iteration. However, the amount of development work varied across teams. In order to focus on the main development effort, we used 20 work items as a cutoff to determine which teams to include in our analysis. This resulted in data for 16 out of the 26 teams. The 16 teams were distributed across 15 development sites. One of the teams had 3 technical leaders, 9 teams had 2 technical leaders and the remaining 6 teams had one technical leader. The 10 excluded teams had just a few members and represented a very small fraction of the overall development effort (7.2%) and communication activity (0.9%).

Project B

We collected data from a multi-national development organization responsible for producing a complex embedded system for the automotive industry. The collected data covered the 27 months of development activities associated with the 9th version of the product from the beginning of the project in the beginning of 2009 until mid of 2011. Four hundred and seventeen developers located in four development sites distributed across Europe and India participated in the project. Those developers were organized in 39 development teams ranging from 7 and 21 members. Most project members belonged to just one of the teams. In addition, individuals have roles within each team and those roles were formalized in the process manual of the project. The defined roles were "project manager", "sub-project manager", "architect", "component owner", "developer", "tester", "system integrator" and "component integrator". Each team had a person with the "component owner" role and the project had 12 members with the role of "architect". Two of the people had both roles "component owner" and "architect". One team had two people with the "component owner" role. Our analyses used the data from all 39 teams.

The project organized its development effort using quarterly deliveries to the customer. The first 8 quarters constituted preliminary versions of the system while the last quarter corresponded to the production release. The tasks to be performed for each quarter and by each development team were planned at the beginning of each quarter. This development process shares some similarities with the agile approach used in Project A. However, the development organization of Project B did not consider itself as using an agile development process but rather an iterative version of more traditional sequential (or V-model) process.

Identification of Technical Leaders

We define technical leaders are those individuals that have strong technical competence and are responsible for the defining, architecting and/or designing the core technical

¹ IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both

properties of part of a complex system. In addition, those individual, typically, have in-depth knowledge of various parts of a system that represent valuable resources for the rest of the development organization. After performing an initial analysis of the datasets and confirming our understanding of the projects' context via interviews, we consider technical leaders in Project A as those individuals with the "component lead" role and in Project B those individuals with the roles of "component owner" and/or "architect". The people with those roles match the characteristics of a technical leader presented in our definition.

Comparison across Projects

The two studied projects were chosen because they represent two very distinct points across two key dimensions of maturity: product maturity and process maturity. Differences across those dimensions are important aspects to consider because they have been found to influence communication and organizational processes in product development endeavors [22]. Table 1 highlights the key differences where Project A represents the less mature development context while Project B exhibited high levels of product and process maturity. It is worth pointing out that the two dimensions of maturity are not completely orthogonal. In fact, they might be confounded. Such a limitation can be addressed from an experimental design point of view. If the impact of the communication patterns of technical leaders, our main focus, is the same across the two projects, then the confounding effect between product maturity and process maturity does not represent a problem.

Table 1: Comparison between Projects along Product and Process Dimensions of Maturity

	Project A	Project B
Process Maturity	Agile process.	Well-established use of V-model approach.
	Highly geographically distributed.	Very mature org. with more than 23 years developing the product.
	No CMM assessment.	
Product Maturity	Novel Product, first in class.	Market-leader.
	First release of product.	9 th generation of the product.

Description of Measures

We used 4 sources of data for our analyses. First, we collected the software repositories from both projects including version control activity, task and defect tracking reports as well as a wide range of technical properties of the software systems being developed. Second, we also collected information about the project member's location from personnel records and corporate directories. In project B, we also collected 13 months worth of communication activity from the project's instant messaging system. Finally, we conducted semi-structured interviews with individuals who were members of the projects throughout the development, to validate our assumptions about the interpretation of the measures (4 in Project A and 16 in Project B).

As discussed earlier, both projects produced a particular outcome in predefined time periods. In Project A, those time periods were called iterations as it is traditional in agile development processes and lasted 6 to 8 weeks. In Project B, those time periods were called delivery periods and they lasted 3 months. We will refer to these time periods as iterations throughout the rest of paper.

Measuring Delivery Performance

Our unit of analysis is the pair (iteration, team). We measured *Delivery Performance* of an iteration as the number of tasks that were planned for the (iteration, team) pair but were rescheduled to the following iteration because they were not completed. This measure focuses on how well the team members that participated in the iteration were able to execute the planned set of activities as well as capturing aspects of the development process such as adequate planning and adequate handling of unexpected situations. Interviews in both projects confirmed that rescheduling was an adequate measure since it was an integral part of the development process.

Measuring Quality

The quality outcomes of an iteration were assessed following the procedure proposed by Cataldo and Ehrlich [8]. For a given iteration i_t , we first selected all the defects reported in the following iteration i_{t+1} . Our measure of *Quality* captured the number of defects that either impacted components changed during iteration i_t and took 1 or more days to get resolved or impacted components not changed in iteration i_t nor in iteration i_{t+1} . Following the guidelines of [8], we validated this measure by analyzing a random sample of defects corresponding to 10% of all the defects reported across all the iterations (527 defects in Project A and 891 in Project B) and found our measure to be reliable.

Construction of the Communication Network

Both projects used development environments that had features that provide project members with contextualized interaction for work items. These interactions took the form of comments in a work item report and were stored in the software repositories of both projects. Those data allowed us to identify the work items, the individuals that made the comment(s) as well as the content of the comment. We constructed iteration-level communication networks following the procedure proposed by Ehrlich and Cataldo [8, 16]. Since tasks could span multiple iterations, we considered only those comments made in the work item reports that were within the time frame of the focal iteration. The first comment in the work item linked the person that made the comment with the owner of the task at the time the comment was made. If the owner and the commenter were the same person, the link was ignored. For the subsequent comments in the work item report, a link in the communication network between two project members was made if the comments were adjacent in the sequence of comments and they were associated with the task at hand. These comments also resulted in a link between the commenter and the owner of the work item at the time it was made.

We also have instance messaging logs for Project B. In this case, we had 3 raters examine the logs and identify the context of the discussions to link them to specific work item. The linkages between individuals were time-stamped so we only included in a focal interaction those interactions that were time-stamped during the time period covered by the iteration. Due to the volume of the data, we divided the data among the 3 raters, however, they had a 15% overlap in the content coded by them. The raters agreed in the interaction linkages as well as in the linkage of comments to work items in 99.2% of the cases.

Communication Patterns of Technical Leaders

We measured the *Technical Leader Centrality* in the communication networks using the traditional degree centrality network measure. We also measured the directionality of the technical leader communication patterns using in-flow and out-flow degree centrality measures. More specifically, we measured *Ratio Leader's In-Flow vs. Rest of Team* and *Ratio of Leader's Out-Flow vs. Rest of Team* as the ratio between the leader's in- or out-degree centrality in the communication network and the average in- or out-degree centrality of all the other team members. In the case the team had more than one technical leader, we averaged the in-flow/out-flow degree centrality of the leaders. We also computed the *Ratio Leader's Out-Flow/In-Flow* measure that captures the proportion of the technical leader's communication that was out-going. More specifically, the measure was computed as out-flow degree centrality / (out-flow degree centrality + in-flow degree centrality).

Additional Control Factors

Past research work suggests a wide range of factors that may impact outcomes such as performance and quality in software development (e.g. [5, 8, 10, 11, 16, 20]). We organized our control variables into six groups:

Team-level Communication Structure: Cataldo and Ehrlich [8] found that two particular team-level communication structures, hierarchical and small-world patterns, impacted product development team outcomes differently. Given those results, we computed the *Hierarchical Communication Pattern* measure based on the hierarchy metric proposed by Krackhardt [26] and *Small World Communication Pattern* measure as proposed in Uzzi and Spiro [36], following the procedure described in Cataldo and Ehrlich [8].

Controls for Volume of Communication-related factors: Since our independent measures focus on patterns of communication, it is important to control for factors related to the propensity of the team members and, in particular, technical leaders to communicate. We computed the *Amount of Intra-Team Communication* as the number of dyads from the communication networks where both members belonged to the same team. On the other hand, *Amount of Inter-Team Communication* captured the number of dyads where at least one of the persons in the pair also belonged to a different team. We also computed the *Leader-to-out-team Communication* measure as the amount of times the

technical leaders interacted with members from other teams. Finally, the technical leaders in both projects were active in the development effort so we measured the propensity of the individuals to comment in work items not owned by them (*Leader Involvement in non-Owned Work Items*) as the number of work items not owned by the technical leader normalized by the total number of work items in the team and team size.

Controls for product-related factors: Technical dependencies that exist among the components of a system are an important driver of work dependencies and consequently, of interaction among interdependent project members (e.g. [12, 33]). Building on the work of Cataldo and Nambiar [11], we computed the *Internal Technical Coupling* measure as the density of the sub-matrix based on the set of relationships defined in the technical dependencies matrix that are contained within a team's area of responsibility. We also computed the *External Technical Coupling* measure as the number of components not under the responsibility of the focal team that had technical dependencies with those components changed by the focal team.

Controls for task-related factors: Since iterations vary in the amount of work they generate and in the importance of the work, we measured the *Ratio of High Priority Work Items* that each team faced in each iteration as well as the *Average Size of Work Items* as the average number of files modified in the commits associated with the work items.

Controls for team-related factors: We computed several measures that capture attributes of the teams we studied. Since comments on work items could also come from people outside the team, we computed two measures: *Number of Project Members Involved* captures the total of team and non-team members that participated in the work associated with each pair (iteration, team) and *Number of Multi-Team Project Members* captures the subset of those individuals that were part of more than one formal team. Both projects involved members across multiple geographical locations in different countries. Therefore, we assessed the geographical dispersion of the teams, *GSD*, as a dichotomous variable where 1 indicates that the project members involved in the work associated with a pair (iteration, team) were geographically distributed; otherwise it was set to 0. We collected measures of experience based on the work by Boh and colleagues [6]. We measured: *Average Level of Technical Experience* as the average number of work items that the team member worked on prior to the focal iteration, *Average Level of Shared-work Experience* measures the average number of times that team members worked together in work items prior to the beginning of the focal iteration, and *Average Level of Team Workload* as the average number of work items owned per team member during the iteration.

Controls for unobserved effects: Finally, we included additional controls for unobserved, team- and iteration-specific factors. These were recorded as Dummy A through F for

the teams (16 teams in Project A and 39 in Project B) and Dummy Iteration A through D for the iterations (14 iterations in Project A and 9 in Project B).

Finally, it is worth pointing out that we computed and evaluated other network measures such as density, which did not have any impact in our models. Therefore, we did not include them in the final results reported in the paper.

Description of the Model

In our analyses, we used a multi-level Negative Binomial regression model because our two dependent variables, *Delivery Performance* and *Quality*, are count variables. We examined our research questions by creating several models, which included the different control variables as well as our independent measures as fixed effects. We also added iteration and team as random effects. In order to assess the fit of each model, we report the log-likelihood of each model as well as the percentage of deviance explained by the model. The percentage of the deviance explained is a ratio of the deviance of the null model (containing only the intercept) and the deviance of the final model. Although coefficients are often reported as part of the results, we opted for reporting the incidence rate ratios (IRRs) associated with the fixed effect factors of the regression models because they simplify the interpretation of the results. IRRs indicate the rate at which the dependent variable would change for one unit change in one independent variable keeping the other factors constant.

RESULTS

We report the results of the various analyses performed. We started our analyses with an examination of issues of collinearity. Secondly, we replicated the results from Cataldo and Ehrlich [8] using new data from Project B. This is an important step because we are examining whether effects of leader position occur even when we account for effects of communication structure in the teams. We conclude with results related to our research questions.

Initial Analyses

As discussed earlier, in Project A we only considered data from 16 out of the 26 teams. The 14 iterations and 16 teams resulted in 197 pairs <iteration, team>. We did not have 224 pairs because not all the teams had development activity planned in all iterations. In the case of Project B, the 39 teams and 9 iterations resulted in 324 pairs <iteration, team>. Again, we did not have 351 pairs because not all teams had planned development activity in all iterations.

Testing for Collinearity

We computed descriptive statistics of the measures described in the previous section and log-transformed all those variables that exhibited skewed distributions. We also performed various collinearity diagnostics. A variance inflation factors (VIF) analysis identified collinearity problems with *Technical Leader Centrality* and *Amount of Intra-Team Communication* measures. We kept the latter measure out of our regression analyses. We only included *Technical Leader Centrality* in a subset of the models. All

the other measures included in our models did not exhibit collinearity problems since they had VIFs below 5 as recommended guidelines suggest [21]. A pair-wise correlation analysis among those measures also showed levels of correlation that were not problematic (all below 0.40).

Replication Analyses

The first step in our analyses consisted in replicating the results of Cataldo and Ehrlich [8] using new data from Project B. Overall, our analyses showed that project B exhibited results consistent with those reported by Cataldo and Ehrlich [8] in terms of higher levels of hierarchical communication patterns being associated with higher levels of delivery performance (IRR: 0.247, $p < 0.01$, log-likelihood = -395.47, deviance explained = 59.3%). In fact, the effect size of *Hierarchical Communication Pattern* is significantly higher in Project B as suggested by the difference in the deviance explained between the full model (59.3%) and the model without the factor (32.6%). On the other hand, we did not find evidence in Project B for an impact of *Small-World Communication Pattern* on delivery performance. It is worth pointing out that Cataldo and Ehrlich [8] found a marginal effect in Project A. We also found that hierarchical communication structures have a significant detrimental effect on quality (IRR: 1.322, $p < 0.01$, log-likelihood = -759.41, deviance explained = 46.5%) while small-world communication structures improved the quality outcomes of an iteration (IRR: 0.393, $p < 0.01$, log-likelihood = -619.14, deviance explained = 56.4%). These results confirm the previous findings and provide the foundation for examining additional effects associated with leader communication.

Technical Leaders and their Centrality in the Communication Patterns of Teams

Our first set of research questions (RQ1a and 1b) concerned the role of centrality of the technical leader. As Table 2 shows, *Technical Leader Centrality* has statistically significant IRRs and those values are between 0 and 1 in all cases. Those results indicate that the more central the technical leader is in the communication network of the team, the more likely the team is to exhibit higher levels of delivery and quality performance. Due to space constraints, we only show a subset of the regression models that we constructed. The impact of *Technical Leader Centrality* was consistent across both datasets and across all models. For instance, the same significance and directionality of impact was found when *Hierarchical Comm. Pattern* was included in the quality performance models or *Small-World Comm. Pattern* was included in the delivery performance models.

Certainly this result is not novel. Numerous studies have highlighted the team-level benefits of centrally positioned leaders (e.g. [3]). Interestingly and despite finding an impact of centrality, further analysis revealed that the leaders were not the most central individuals in their teams suggesting that technical leaders need to be engaged in the flow of communication but do not need to dominate or control the flow. Figure 1 shows aggregate average values of degree centrality (DC) across both projects for those teams that

were in the lower quarter in both outcome measures as well as those teams in the upper quarter on both measures. We observe that level of centrality of the technical leaders was marginally above the average centrality level across the other team members. Furthermore, an examination of each pair (iteration, team) from both projects revealed that in only 7 out of the 521 pairs, the technical leader was the team member with the highest degree centrality level. In 378 pairs technical leaders were in the third quartile (considering the forth quartile as the highest). In the remaining pairs (136), the leaders were positioned in the lowest 2 quartiles.

Figure 1: Aggregate Average Levels of Centrality
Quality

		Low	High
Delivery	Low	Leader DC=0.032 (0.077) Rest DC=0.027 (0.069)	
	High		Leader DC=0.137 (0.248) Rest DC=0.107 (0.221)

Technical Leaders and the Impact of the Directionality of their Communication Patterns

The next step in our analysis was to examine how the directionality of the communication of technical leaders impacts the teams' performance (Research questions 2a, 2b, 3a an3b). Tables 3a, 3b and 3c report the results of our regression analyses. It is important to point out that in those three tables, we only report the odds ratio of our independent measures *Ratio Leader's Out-Flow/In-Flow*, *Ratio Leader's In-Flow vs. Rest of Team* and *Ratio of Leader's Out-Flow vs. Rest of Team*, respectively, and the fit of the various models. The *Ratio Leader's Out-Flow/In-Flow* measure has a consistent positive effect on performance across both projects. The higher the amount of out-flow degree centrality the technical leader exhibits in his/her communication patterns relative to the in-flow degree centrality, the better the delivery performance (the number of rescheduled tasks are reduced) and quality performance (the number of defects is reduced) of the teams as shown by the statistically significant odds ratios below 1 in Table 3a. Comparing the fit of the models in Table 2 and Table 3a, we observe that considering the ratio of the out-/in-degree centrality has a significantly large size effect in the models, averaging about 5% more deviance explained in Project B, about 14% more in project A in the delivery performance models and about 7% more in Project A in the quality performance models. Combined, these results suggest that teams benefits when leaders provide more information than they receive.

Arguably, the impact of out-flow communication could suggest that the benefits stem from leaders being directive rather than sharing information. However, an examination of a random sample of interactions in both projects provided some valuable insights. Overall, technical leaders tended to engage mostly in interactions that answer questions, provide suggestions about technical challenges, pro-

vide advice about the best course of action and provide pros/cons analysis of proposed ways to tackle a technical problem. For example, we identified a pair of technical leaders with similar communication network metrics (Leader #1, In-degree Centrality = 0.0309, Out-degree Centrality = 0.0451 – Leader #2, In-degree Centrality = 0.0287, Out-degree Centrality = 0.0575) but from two different teams to explore whether there might be additional effects of communication style that could be explored in future work.. We identified very distinct styles between those leaders. Leader #1 tends to be directive in their interactions as this typical example shows: “...*The problem here is the XUL runner which tweaks the selection. Have to investigate how we deal with links in the edit mode anyway...*”. On the other hand, leader #2 seems to be more constructive in their interactions “...*My reading of the material suggests that it should be: it's a secondary command (a peripheral action that, while helpful, isn't essential to the purpose of the window); it navigates to another page, window, or Help topic; and it's label is plenty more than recommended four word limit. I ask not ... to see whether we end up at the same point after reading the same material. If you and I can't agree - with all the additional background material we have available - it's going to be tough to get the regular developer on board (when they're not going to have the time or inclination to wade through the pages of content that we've waded through)...*”.

In Tables 3b and 3c, we report the results of our analyses of directionality of communication patterns addressing research questions 2a and 2b (in-flow of communication). We compare the amount of in-flow (Table 3b) or out-flow (Table 3c) communication from the technical leaders relative to the rest of the team. Our analyses do not provide evidence that *Ratio Leader's In-Flow vs. Rest of Team* influences either performance outcome in the teams (see Table 3b). On the other hand, Table 3c shows that the *Ratio Leader's Out-Flow vs. Rest of Team* has a consistent positive impact across both projects and across both performance outcomes.

These results further suggest that the directionality of the communication of technical leaders is an important factor to consider in terms of two key product development outcomes: delivery and quality performance.

DISCUSSION

Formal leaders play an important role in the functioning and well-being of teams [3, 4, 28]. However, the increasing prevalence of self-organized teams [15], emergent leaders [39] and shared leadership [41], raises the question of whether formal leaders' have a significant effect on a team's performance. We addressed this question by studying the pattern of communication between leaders and team members in 55 product development teams from two companies. Specifically, we examined whether the leader's position in the team's communication network and the direction of their communication was associated with team

Table 2: The Impact of “Technical Leaders Centrality” in the Team Communication Network

	Delivery Performance				Quality Performance			
	Project A		Project B		Project A		Project B	
	Model I	Model II	Model III	Model IV	Model V	Model VI	Model VII	Model VIII
Controls for Unobserved Effects								
<i>Team Dummy A</i>	1.171*	1.402*	1.403**	1.305**	1.079*	1.102*	1.109**	1.141**
<i>Team Dummy B</i>	0.723**	0.899**	0.991	1.092	0.872**	0.798**	1.011	0.929
<i>Team Dummy C</i>	1.011	0.941	0.801**	1.089**	0.589	0.945	1.201**	1.011
<i>Team Dummy D</i>	1.429**	0.899**	1.195*	0.901*	0.621**	0.749**	0.848*	0.851*
<i>Team Dummy E</i>	-	-	1.178*	0.937*	-	-	0.561*	0.374*
<i>Team Dummy F</i>	-	-	0.684*	1.099*	-	-	1.071*	0.927*
<i>Iteration Dummy A</i>	0.601**	0.602**	0.402	0.389	0.903**	1.202**	1.017	1.189
<i>Iteration Dummy B</i>	0.257**	0.478**	0.501	1.745	0.486**	0.382**	0.809	0.486
<i>Iteration Dummy C</i>	0.071**	0.079**	0.152*	0.392*	0.078**	0.092**	0.121*	0.091*
<i>Iteration Dummy D</i>	1.191*	1.414*	1.892**	1.451**	1.109*	1.203*	1.055**	1.203*
Controls for Team-related factors								
<i>Num. of Members</i>	6.249**	8.194**	9.204**	4.244**	1.893**	1.943**	2.422**	2.892**
<i>Num. of Multi-team Members</i>	0.392*	0.674*	0.902*	0.934*	1.023	1.294	1.349	1.195
<i>GSD</i>	1.485**	1.568**	2.044**	1.605**	1.478**	1.345**	1.882**	1.922**
<i>Avg. of Tech. Experience</i>	0.795*	0.985*	0.678**	0.895**	0.899*	0.892**	0.809**	0.985**
<i>Avg. of Shared-work Exp.</i>	0.506	0.793	0.793*	0.673*	1.102	1.185	0.692	0.902
<i>Avg. of Team Workload</i>	1.356**	1.545**	1.935**	1.932**	2.119**	1.902**	1.242**	1.155**
Controls for Task-related factors								
<i>Ratio of High Priority Tasks</i>	0.342**	0.842**	0.724*	0.799*	0.702**	0.678**	0.289**	0.678**
<i>Average Size of Tasks</i>	1.055	1.002	0.899	0.994	1.894**	1.792**	0.992	0.995
Controls for Product-related factors								
<i>Internal Tech. Coupling</i>	1.485*	1.593*	3.842**	4.522**	2.195**	1.953**	3.092**	2.827**
<i>External Tech. Coupling</i>	1.249**	1.345**	1.784**	1.902**	1.154**	1.274**	1.986**	1.752**
Controls for Volume of Comm.								
<i>Intra-team Comm.</i>	0.895*	0.951*	0.824*	0.842*	0.930*	0.917*	0.792*	0.801*
<i>Leader-to-out-team Comm.</i>	1.795**	2.294**	2.224**	4.002**	1.701**	1.643**	1.195**	1.442**
<i>Leader in not-Owned Tasks</i>	1.143**	1.687**	2.394**	2.032**	1.762**	1.985**	2.249**	1.942**
Comm. Network Structure								
<i>Hierarchical Comm. Pattern</i>	0.394**		0.304**		1.302**		1.509**	
<i>Small World Comm. Pattern</i>		1.785**		1.013		0.892**		0.455**
Leader Comm. Pattern								
<i>Technical Leader Centrality</i>	0.875**	0.911**	0.567**	0.703**	0.689**	0.621**	0.263**	0.301**
Log-Likelihood	-401.87	-404.59	-380.01	-609.74	-468.17	-472.73	-659.89	-561.54
Deviance Explained	40.6%	40.2%	60.9%	37.2%	54.0%	53.5%	53.5%	60.4%

(+ p < 0.10, * p < 0.05, ** p < 0.01)

performance measures of productivity and quality. Previous research demonstrated that product development teams are more productive when they have a hierarchical communication structure, and, they produce higher quality products when they have a small-world communication structure [8]. This finding, replicated in the present study, says that the team's own communication contributes to performance. The present study examined whether the leader's pattern of

communication has any additional effect on team performance. Several important results emerged.

First, we found that teams performed better when the leader was central in the network (RQ 1a,b) confirming previous research [3, 28]. Surprisingly, the leader was not always the most central in the network, implying that leader centrality may not be as simple as previously thought [29, 34]. One interpretation is that we are tapping into a model in which

people who emerge as central in a network [14] come to share responsibilities with formal leaders [32, 41].

Second, we established that teams had higher productivity and higher quality when leaders communicated out to the team (RQ 3a,b) even after accounting for the significant effects of hierarchical and small-world communication structures. This is an important finding because it provides strong empirical evidence that leaders not only contribute to the general well-being of a team but their position in the communication network has a measurable effect on the team's performance. This result is consistent with leader-member exchange theory, which emphasizes the importance of relational behaviors between leaders and members [4, 39]. The present study contributes to and extends that line of research by demonstrating the importance of leader centrality within the out-flow communication.

Third, we did not find any benefit for either productivity or quality for teams whose leaders were receivers of information (RQ 2a,b). Leaders who are centrally positioned within the in-flow of communication are often perceived as more important or having higher expertise [28, 40]. In a technical environment, expertise is highly valued so it is surprising that we did not find effects of perceived expertise on team performance. Perhaps leader expertise is less important than the perceived expertise of team members.

Even though product development teams can succeed without leaders [15], a key contribution of this study is to demonstrate that technical leaders are important to the team's performance even after accounting for the team's communication. Moreover, the replication of our results across two distinct companies speaks to the generality of our findings.

Limitations

Our study has some limitations worth pointing out. We relied on archival data from asynchronous communication which under-reports any face-to-face communication that might have occurred between collocated team members. Interviews with some of the members of Project A confirmed that developers, even those who were collocated, were encouraged to discuss technical issues in the comments field that we used as our data source, so that issues could be tracked and be visible to all developers regardless of their location. A similar finding was obtained through interviews in Project B. While we obtained strong results with respect to the structure and flow of communication in product development teams, there is value in studying the content of the communication. A complete analysis of the content is beyond the scope of this paper. However, a very preliminary comparison between leaders with similar network metrics revealed marked differences with one leader having a directive style and the other leader adopting a more constructive and conversational style. Future research using content analysis could shed light on additional aspects of the communication of technical leaders.

Implications for New Tools

The results of the present study suggest that leaders might benefit from tools to adjust their communication. First, there is a direct implication from the growing body of work in the area of communication, coordination and awareness tools. For example, tools that support task-centric interactions such as task or defect tracking systems (e.g. Bugzilla) or integrated development environments (e.g. IBM RTC) could be enhanced to provide concrete guidance to technical leaders regarding their in/out flow communication patterns. Furthermore, tools such as Worldview [31] or Ensemble [38] could be enhanced to support the identification of dependencies where technical leaders could play a role in working with those most directly affected.

A different breed of tools comes from research, which used visual feedback to improve team performance [25, 30]. Kim and Pentland [25] used sociometric badges to capture group dynamics in real-time and display the results to each team member. They obtained a significant change in behavior for both collocated and distributed groups who received visual feedback on group dynamics. In particular, there were fewer overlapping conversations, greater interactivity, less dominance and distributed teams became more like collocated teams. A similar approach could be developed to provide feedback to leaders.

A third approach can be found in recent research on leaders of online communities. Matthews and colleagues [27] developed a set of metrics reflecting the participation and engagement of community members over time that leaders could use to intervene in the community. What was especially valuable to leaders were metrics that helped them identify critical contributors and content. A similar approach in product development teams could help leaders identify which people to communicate with to have the greatest impact on the team.

CONCLUSION

The contributions of our study are two-fold. First, our analyses and results linked the leader position in the team's communication network and in the flow of communication to measurable performance outcomes of product development teams. Finally, our results were obtained after controlling for other structural effects of communication on team performance and replicated in two companies with very distinct product and process maturity.

ACKNOWLEDGMENTS

This research was partially sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Table 3a: The Impact of the “Ratio Leader’s Out-Flow/In-Flow” Communication Patterns

	Delivery Performance				Quality Performance			
	Project A		Project B		Project A		Project B	
	Model I	Model II	Model III	Model IV	Model V	Model VI	Model VII	Model VIII
Comm. Network Structure								
<i>Hierarchical Comm. Pattern</i>	0.436**		0.281**		1.411**		1.497**	
<i>Small World Comm. Pattern</i>		1.993**		1.001		0.879**		0.478**
Leader Comm. Pattern								
<i>Ratio Leader’s Out-Flow/In-Flow</i>	0.899**	0.878**	0.886**	0.871**	0.272**	0.257**	0.271***	0.326**
Log-Likelihood	-306.27	-309.35	-335.36	-564.98	-422.37	-383.92	-593.7	-495.26
Deviance Explained	54.8%	54.3%	65.5%	41.8%	58.5%	62.3%	58.2%	65.1%

(+ p < 0.10, * p < 0.05, ** p < 0.01)

Table 3b: The Impact of the “Ratio Leader’s In-Flow vs. Rest of Team” Communication Patterns

	Delivery Performance				Quality Performance			
	Project A		Project B		Project A		Project B	
	Model I	Model II	Model III	Model IV	Model V	Model VI	Model VII	Model VIII
Comm. Network Structure								
<i>Hierarchical Comm. Pattern</i>	0.427**		0.249**		1.219**		1.392**	
<i>Small World Comm. Pattern</i>		1.916**		1.003		0.929**		0.401**
Leader Comm. Pattern								
<i>Ratio Leader’s In-Flow vs. Rest of Team</i>	0.948	0.923	0.945	0.927	0.967	0.929	0.792+	0.825+
Log-Likelihood	-414.19	-420.07	-394.69	-645.91	-489.71	-498.01	-747.09	-607.99
Deviance Explained	38.8%	37.9%	59.3%	33.5%	51.9%	51.1%	47.4%	57.2%

(+ p < 0.10, * p < 0.05, ** p < 0.01)

Table 3c: The Impact of the “Ratio Leader’s Out-Flow vs. Rest of Team” Communication Patterns

	Delivery Performance				Quality Performance			
	Project A		Project B		Project A		Project B	
	Model I	Model II	Model III	Model IV	Model V	Model VI	Model VII	Model VIII
Comm. Network Structure								
<i>Hierarchical Comm. Pattern</i>	0.507**		0.465**		1.989**		1.983**	
<i>Small World Comm. Pattern</i>		1.804**		0.999		0.906**		0.298**
Leader Comm. Pattern								
<i>Ratio Leader’s Out-Flow vs. Rest of Team</i>	0.901**	0.793*	0.833*	0.901*	0.985	0.755*	0.704**	0.561*
Log-Likelihood	-349.19	-397.72	-378.52	-618.13	-495.25	-483.41	-698.17	-597.69
Deviance Explained	48.4%	41.2%	61.0%	36.3%	51.3%	52.5%	50.8%	57.9%

(+ p < 0.10, * p < 0.05, ** p < 0.01)

REFERENCES

1. Ancona, D. and Caldwell, D. Bridging the boundary: External activity and performance in organizational teams. *Administrative Science Quarterly*, 37 (1992), 634-665.
2. Ancona, D. G. and Bresman, H., *X-teams: How to build teams that lead, innovate, and succeed*: Harvard Business Press, 2007.
3. Balkundi, P. and Harrison, D. A.. Ties, Leaders, and Time in Teams: Strong inference about network structure's effects on team viability and performance. *Academy of Management Journal*, 49 (2006), 49-68.

4. Balkundi, P. and Kilduff, M.. The ties that lead: A social network approach to leadership. *The Leadership Quarterly*, 17 (2006), 419-439.
5. Bass, M., *et al.*. Architectural Misalignment: An Experience Report. In *Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture* (2007).
6. Boh, W. F., *et al.*. Learning from Experience in Software Development: A Multilevel Analysis. *Management Science*, 53 (2007), 1315-1331.
7. Burt, R. *Brokerage and closure: An introduction to social capital*. Oxford University Press, USA, 2005.
8. Cataldo, M. and Ehrlich, K. The impact of communication structure on new product development outcomes. In *Proc. CHI 2012*, ACM Press (2012).
9. Cataldo, M. and Herbsleb, J. D. Communication networks in geographically distributed software development. In *Proc. CSCW 2008*, ACM Press (2008).
10. Cataldo, M. and Herbsleb, J. D. Coordination Breakdowns and Their Impact on Development Productivity and Software Failures., *IEEE Transactions on Software Engineering*, 39 (2013), 343-360.
11. Cataldo, M. and Nambiar, S. The impact of geographic distribution and the nature of technical coupling on the quality of global software development projects. *Journal of Software Maintenance and Evolution: Research and Practice*, 24 (2012), 153-168.
12. Cataldo, M., *et al.* Identification of coordination requirements: implications for the Design of collaboration and awareness tools. In *Proc. CSCW 2006*, ACM Press (2006).
13. Cramton, C. The mutual knowledge problem and its consequences for dispersed collaboration. *Organization Science*, 12 (2001), 346-371.
14. Cross, R., *et al.* Supporting knowledge creation and sharing in social networks. *Organizational Dynamics*, 30 (2001), 100-120.
15. Crowston, K., *et al.* Self-organization of teams for free/libre open source software development. *Information and Software Technology*, 49 (2007), 564-575.
16. Ehrlich, K. and Cataldo, M. All-for-one and one-for-all?: a multi-level analysis of communication patterns and individual performance in geographically distributed software development. In *Proc. CSCW 2012*, ACM Press (2012), 945-954.
17. Espinosa, J. A., *et al.* Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development. *Organization Science*, 18 (2007), 613-630.
18. Faraj, S. and Sambamurthy, V. Leadership of information systems development projects. *IEEE Transactions on Engineering Management*, 53 (2006), 238-249.
19. Faraj, S. and Sproull, L. Coordinating Expertise in Software Development Teams. *Management Science*, 46 (2000), 1554-1568.
20. Graves, T. L., *et al.* Predicting fault incidence using software change history. *IEEE Transactions on Software Engineering*, 26 (2000), 653-661.
21. Hair, J. F., *et al.* *Multivariate data analysis 5th Edition*. Upper Saddle River, NJ: Prentice Hall, 1998.
22. Henderson, R. M. and Clark, K. B. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, 35 (1990), 9-30.
23. Herbsleb, J. D. and Mockus, A. An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Trans. Softw. Eng.*, 29 (2003), 481-494.
24. Hossain, L., *et al.* Actor centrality correlates to project based coordination. In *Proc. CSCW 2006*, ACM Press (2006), 363-372.
25. Kim, T., *et al.* Meeting mediator: enhancing group collaboration using sociometric feedback. In *Proc. CSCW 2008*, ACM Press (2008), 457-466.
26. Krackhardt, D. Graph theoretical dimensions of informal organizations. In *Computational Organization Theory*, Psychology Press (1994), 89-111.
27. Matthews, T., *et al.* Community insights: helping community leaders enhance the value of enterprise online communities. In *Proc. CHI 2013*, ACM Press 2013, 513-522.
28. Mehra, A., *et al.* The social network ties of group leaders: Implications for group performance and leader reputation. *Organization Science*, 17 (2006), 64-79.
29. Mehra, A., *et al.* The social networks of high and low self-monitors: Implications for workplace performance. *Administrative Science Quarterly*, 46 (2001), 121-146.
30. Pentland, A. S., *et al.*, Awareness as an antidote to distance: Making distributed groups cooperative and consistent. In *Proc. CSCW 2012*, ACM Press 2012, 1237-1246.
31. Ripley, R. M., *et al.* A Visualization for Software Project Awareness and Evolution. In *Proc. VISSOFT 2007*. IEEE Press 2007, 137-144.
32. Robert, L. P. A multi-level analysis of the impact of shared leadership in diverse virtual teams. In *Proc. CSCW 2013*, ACM Press 2013, 363-374.
33. Sosa, M. E., *et al.* The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science*, 50 (2004), 1674-1689.

34. Sparrowe, R., *et al.* Social networks and the performance of individuals and groups. *Academy of Management Journal*, 44 (2001), 316-325.
35. Ulrich, K. T. and Eppinger, S. *Product Design and Development 4th Edition*. New York, NY: McGraw-Hill (2007).
36. Uzzi, B. and Spiro, J. Collaboration and creativity: The small world problem. *American Journal of Sociology*, 111 (2005), 447-504.
37. Weisband, S. P. *Leadership at a distance: research in technologically-supported work*. Psychology Press (2008).
38. Xiang, P. F., *et al.* Ensemble: a recommendation tool for promoting communication in software teams,. In *Proceedings of the Int'l Workshop on Recommendation Systems for Software Engineering*, IEEE Press 2008.
39. Yoo, Y. and Alavi, M. Emergent leadership in virtual teams: what do emergent leaders do? *Information and Organization*, 14 (2004), 27-58.
40. Yukl, G., *et al.* Influence of leader behaviors on the leader-member exchange relationship. *Journal of Managerial Psychology*, 24 (2009), 289-299.
41. Zhu, H., *et al.* Effectiveness of shared leadership in online communities. in *Proc. CSCW 2012*, ACM Press 2012, 407-416.