The Impact of Collaboration Network Structure on Issue Tracking’s Process Efficiency at a Large Business Software Vendor

Arne Beckhaus, Lars M. Karg
SAP Research
Bleichstr. 8
64283 Darmstadt, Germany
{arne.beckhaus, lars.karg}@sap.com

Dirk Neumann
University of Freiburg
Platz der Alten Synagoge, KG II
79085 Freiburg, Germany
dirk.neumann@is.uni-freiburg.de

Abstract

Researchers in the IS domain have addressed communication structure and its effect on performance. While early research focused on small networks and utilized sociometric surveys, recent works have concentrated on electronic data sources provided by open source software repositories. Surprisingly, software vendors have less frequently been studied despite their need for continuous enhancement of organizational design. In this study, we analyze a global business software vendor by utilizing existing data sources. We investigate the association of both collaboration network centrality and communication with process efficiency. Despite coping with complex, interwoven processes and social networks, we find that communication, centrality in the large case-study-wide network, and communication pattern homogeneity are positively associated with process efficiency. However, centrality in small work groups slows the analyzed issue tracking process down, possibly due to increased overhead and bottleneck effects that become visible when looking at single issues qualitatively.

1. Introduction

Many companies in the software sector are only few decades of age and have grown at fast pace. This growth has not been limited to revenues but has been accompanied by constant hiring and expanding organizational structures. This environment poses demanding challenges to the organization of hierarchy and processes. Our research’s objective is to enhance the transparency of the effects influencing performance in this complex sociologic and organizational setup.

Researchers in Information Systems (IS) have recently developed increased interest in social network analysis (SNA) (cf. [1]). This methodology is often used to study the effect of various aspects of social structure on efficiency, productivity, or other performance measures. Existing research typically analyzes either open source software (OSS) repositories (e.g. [2], [3]) or stems from academic experiments (e.g. [4], [5]). Though being related from a user perspective, open and closed source software are perceived as two largely contrary concepts in literature, not only in terms of their innovation model (e.g. [6]) but also in their development model (e.g. [7]). There are plenty of studies devoted to open source software that share our research objective as stated above. However, our motivation for an understanding of performance influences of organizational design in software development is of even higher interest to large software vendors. They strive for lean processes and profitable software engineering while at the same time having more control over the organizational design of their operations than the OSS community with its voluntary character.

Our research question is whether collaboration network structure is associated with issue tracking efficiency at large business software vendors. This research question has not yet been addressed in literature despite its manifold implications for the management of software companies.

Since the availability of holistic data about the software engineering process cannot be taken for granted, we focus on software quality assurance. It is typically digitized in form of issue tracking systems and has the advantage to be one of the few processes in software engineering that the entire development organization is involved in (c.f. [8]). Related work confirms that this restriction in scope is not too narrow but common
practice in the more advanced research field of OSS (e.g. [9]).

Our paper is organized as follows. In Section 2, we introduce our conceptual background. In Section 3, we explain the research methodology. A qualitative, bottom-up social network analysis of sample issues is presented in Section 4. Its structural findings are used for the design of a subsequent quantitative study in Section 5 in order to empirically validate our research question on the entire dataset. We close our paper with a discussion and the conclusions in Section 6.

2. Conceptual Background

Coordination has been a vital research topic in the field of software development for decades (e.g. [10], [11]). Its importance is best emphasized by Conway’s law [10] which states that the structure of software systems mirrors the structure of their software development organizations. Since software development processes are human-centered, they require a high degree of collaboration [12].

While an efficient process in the manufacturing industry is one with reliable automation and little manual interference, the software industry has not reached the same degree of process maturity yet [13, p. 5]). Processes in the software industry must rather be seen as human centered and collaborative. We assume that communication is positively associated with process efficiency since communication is a central aspect of collaboration. When knowledge workers exchange information frequently and consult peers for expertise, processes in the software industry are assumed to be more efficient. Bulkley and van Alstyne found that more frequent communication has a positive influence on process output [14].

Closely related to communication and collaboration in software development processes is the way the executing teams are familiar with the process. If every iteration of a more or less standardized process is perceived as a new work situation and is handled differently, the process is not mature enough yet. Mapped to the collaborative software industry processes, we are interested in the communication patterns over multiple process iterations. When communication in execution of collaborative processes is heterogeneous, we assume that the process might not be established yet and that a negative influence on performance can be observed. Gloor et al. empirically confirm that balanced communication behavior and performance are associated [15].

Since communication and network topologies do not always depend on each other [14], we are also interested in the SNA perspective on our research. SNA has been successfully applied in a variety of research domains over the last decades. Starting in the social sciences, the IS domain has recently developed an increased interest (cf. [1]). There are many studies in literature that have related social network structure and performance (e.g. [16], [17]). However, only a few studies deal with corporate data whereof we could not identify a single study that takes place in the commercial software industry. While some studies analyze student collaboration (e.g. [4], [5]), many others deal with open source data (e.g. [2], [3]). Though being an interesting and readily available data source, the studies of OSS data have a very broad focus and often analyze hypotheses not transferable to a business context. For example, it is frequently investigated whether the personal networks of project members are helpful in the acquisition of new voluntary team members (e.g. [18]). However, this research question is limited to the OSS domain and not transferable to the software industry lacking the voluntary character of OSS. In commercial software development projects employees are assigned to projects by managers.

Independent of the focus of the study, centrality—as a metric expressing the importance of individuals within a network [19]—has been found to be an important concept of SNA. Barbagallo et al. [20] state that centrality "recalls notions like social power, influence, and prestige." The association between centrality and different facets of performance have been frequently analyzed (e.g. [21]). Centrality has been analyzed on both small and large networks, though yielding more expressive results when applied in the large [22].

Closely related with coordination and the difference between small and large collaboration networks is the concept of economies of scale. These have been confirmed to exist in software development [23]. Our research question therefore extends to the analysis whether large groups of employees executing a process leverage these effects and therefore influence process efficiency.

3. Research Methodology

We conduct our research in an industrial setup at a global player in the enterprise resource planning (ERP) software market. Despite not being able to explicitly name product groups for confidentiality reasons, we describe the research setting and justify our focus on the software quality assurance process. Subsequently, our data collection based on an issue tracking system is described.
3.1. Research Setting

We conducted our study at SAP, a global player in the enterprise resource planning (ERP) software business. The company had over 50,000 employees and revenues of over 15 billion US-dollars in 2008. We cooperated with one department in this company that is organized according to state of the art software development practices as agile development (cf. [24]) and technologically spans three architectural layers.

3.2. Data Collection

In literature, two alternative data acquisition strategies have predominately been used. On the one hand, researchers used questionnaires to conduct sociometric studies (e.g. [4], [5], [12], [17], [25], [26]). They typically capture networks of small size since their respondents must make ties with peers explicit. When there is a reasonable overlap between distinguishable groups that can be surveyed for sociometric data and units of managerial interest, this data acquisition strategy is well suited. It also enables researchers to gather additional variables like performance measures, though bearing the risk of endogeneity by deriving social performance evaluations (cf. [27]).

On the other hand, researchers have utilized network data coded in some electronic form (e.g. [2], [14], [20]). They often rely on publicly available data from open source software repositories. While open source has achieved market legitimacy, the development model significantly differs from proprietary software organizations (e.g. [6], [28]). In a less distributed and more connected development organization that is also strategically aligned for profitable software development, data availability is a greater problem than in open source development communities. Nevertheless, we find that the utilization of existing data bases for derivation of social networks is the more appropriate data acquisition methodology. Software development is a complex socio-technical activity [29] of high-end knowledge work that is typically managed more effectively by fostering collaboration than by defining a strict organizational group structure [30]. The agile process development present in our case study is accompanied by complex relations between employees that frequently cross organizational boundaries like work groups or projects. There is no such concept of a group of employees active in the core software development processes that are not highly interconnected with other groups. The option of a sociometric survey among all members of a well separated group is therefore not feasible in our setup.

We find that the quality assurance process is best suited for network data extraction. This is primarily due to the existence of a uniform issue tracking system used across all departments. It also fulfills the demand to not only host network data but also consistent organizational attributes like process performance. We are aware that the issue tracking system cannot be enriched by other development systems, as in open source, but still judge it to be the best and most complete data source available.

The proposed research setup has the advantage that utilization of existing data sources is non-intrusive to the studied organization. More importantly, [17] point out that social structure has often been analyzed in terms of formal relationships and not by assessing informal communication patterns. This is surprising, since a mixture of formal and informal workgroups is typically used in organizations in order to conduct essential business tasks [26]. Additionally, [31] found that social networks influence efficiency most when people work in a process consisting of search and deliberation. Since issue tracking in an industrial setup is less a matter of searching for a technical solution but for colleagues who know the answer, it matches this description.

The issue tracking database analyzed in this case study is unique and has not been considered by other authors yet. Our corpus of issues comprises 60 months of internal issue tracking data conducted in the analyzed application group. We have pre-processed these data by removing unresolved, i.e. currently ongoing issues. Therefore, we are able to calculate an ex-post performance measure for all 16,782 issues included in our study.

3.3. Case Study Design

Our research question of the impact of communication and social network structure on process efficiency in software quality assurance poses several challenges to the case study design. On the one hand, social network analysis is typically carried out in a bottom-up approach in order to understand the data on a detailed level. We therefore conduct a qualitative study of our data and the networks that can be derived from it in Section 4. On the other hand, our industrial data set which stems from digital data sources is a lot bigger than typical SNA datasets that stem from sociometric surveys. Since we cannot investigate all details qualitatively, we further investigate our structural findings empirically. In Section 5 we statistically analyze the association between social network structure and process efficiency.
4. Qualitative Study

4.1. Analysis from Issue to Global Collaboration Networks

The focus of our study is on the issue tracking process of a department within SAP. Of the data fields contained in the issue tracking system, we are primarily interested in issue reports, the software components that issues are assigned to, the employees that participated in the resolution of the issue, and the order of the actions performed during this resolution. Using these data, we are able to construct social networks of all employees collaborating in the resolution of issues of a particular software component. We have illustrated this chain of analysis in Figure 1.

Part (a) shows the processing chain of an example issue. Four users were involved in the resolution of the problem. User 1 initially reported the message which was first processed by User 2 who needed to re-contact the initial author for some clarifications on how to reproduce the problem. After User 1 had provided additional information, User 2 could identify the source of the problem and direct it to a responsible colleague, User 3. User 3 knew the developer who had written the source code of the erroneous module (User 4) and passed the issue to him. User 4 required another round of re-assuring that he understood the problem correctly with User 2 and finally solved the problem. Once solved, the issue was automatically marked as completed and routed to the author who confirmed the correct solution afterwards (both completion and confirmation are illustrated with squares instead of arrow-heads in Figure 1.(a)).

In this example of a typical issue and its processing chain, we can observe several key characteristics of the issue tracking process at our department. First, the illustration reveals that there are certain time lags where the issue is assigned to a user but s/he does not process it right away. In the specific issue illustrated in (a), User 2 has the longest reaction times. The total processing time of an issue is therefore not only determined by the hours in which work on the issue was actually carried out, but also of times of waiting. A second key finding is the structure of the processing chain. In our example, we can observe seven assignments (arrows) between different processors before User 4 finally solves the problem. However, there are only four users affected by these assignments which indicates quite a lot of re-assuring with prior authors. The experts from our cooperating department call this phenomenon 'ping-pong' and state that it is a frequent communication pattern. Analyzing around 40 issues in a qualitative way revealed that some of these patterns are frequent but that other factors like technical difficulties or the global distribution of responsible experts determined the issue cycle times more. We therefore abstracted from the issue-centric perspective of (a) and took a group perspective that allows social network analysis.

In Figure 1.(b) we have illustrated the collaboration network of all employees involved in the tracking of issues of the same software component as that of (a). The four users of the issue shown in (a) are highlighted in red color in (b). A link between two employees is established once they collaborated on the resolution of at least one issue of the specific software component. In this figure, we can observe that the user illustrated at the very bottom was most active in the specific software component. Our analysis revealed that this was User 2 of (a). He happens to have the role to dispatch new issues on this software component to suitable colleagues and was the second author of many of the issues of this component.

While the group-based collaboration network illustration provides some key insights that we will investigate in more detail next, we first introduce the third level of analysis which is illustrated in Figure 1.(c). A clipping of the dataset-wide collaboration network
is illustrated here. In the same way as in Figure 1.(b), two employees are linked when they have collaborated in the resolution of an issue, however this time independent of the issue’s software component. The per-component networks of (b) are particularly interesting in terms of their centralization. In (c), however, we are interested in the average centralities of certain employees within the global collaboration network (e.g. that of the ten employees of (b) within the network of all employees).

4.2. Analysis of Group-based Collaboration Networks

When looking at collaboration in issue tracking, three perspectives can be taken as illustrated in Figure 1 and explained in Section 4.1. The per-issue perspective (a) has the disadvantage of being highly determined by the content of the technical problem described in the issue. Additionally, the high number of issues in our dataset (16,782) makes it impossible to qualitatively investigate a representative sample. The global collaboration network perspective (c) has the disadvantage of being too big and too frequently linked in order to be qualitatively interpretable. It is of higher value to the quantitative analysis presented in Section 5. The second part of our qualitative study therefore focuses on the group-based collaboration network perspective as illustrated in Figure 1.(b).

Out of the 80 software components present in our data set, we have closely studied a representative sample of 10 as illustrated in Figure 2. For each group’s collaboration network, we have counted the number of employees involved in the resolution of issues of the corresponding software component (variable n) and the social network metric of closeness centralization (variable c, cf. [32]). We have identified an interesting variance in both the group sizes and centralizations and therefore clustered accordingly. Group sizes are clustered by rows (small group sizes in the first row ((a)-(e)), large ones in the second row ((f)-(j))) and centralizations are clustered by columns (lowest centralization in the first column ((a) and (f)) and highest centralization in the fifth column ((e) and (j))).

The examples illustrated in Figure 2 reveal several process characteristics. We observe that the collaboration networks differ not only in size and centralization but also in density. Network (b) appears to have the lowest and (h) the highest density. Overall, the ten networks reveal that collaboration in issue tracking differs a lot between groups. While some groups like (g) communicate very frequently but in an uncentralized way, others like (e) and (j) are organized around a central person that accounts for a large part of the entire group’s collaboration. A closer look at the data revealed that in most cases of the highly centralized networks, the most central employees happen to execute the dispatcher role as discussed in Section 4.1.

4.3. Qualitative Findings

In our qualitative study, we identified three levels of analysis of collaboration in the issue tracking process. The software component based collaboration networks
of groups appear to be most suitable. This is not only due to the size of 80 observations, but also due to the qualitatively identified variance in network structure. From our observation of delays in processing by User 2 in Figure 1 and the high centrality of users in each of Figure 2(e) and 2(f), we postulate that centralization in these groups based collaboration networks might introduce a bottleneck since relying on a key user for information exchange makes the entire issue tracking efficiency dependent on his or her availability. Especially the introduction of a dispatcher role, i.e. a central person to re-route new issues to responsible colleagues, showed to often slow the process down for the same bottleneck reasons.

Our qualitative findings suggest that collaboration network centralization determines the issue tracking process to a high degree. We could identify bottleneck situations like the introduction of a dispatcher role for routing of new issues. In order to validate these bottom-up findings from an empirical, top-down perspective, we have conducted an additional quantitative study which will be presented in the next section. It makes use of the process characteristics identified in the qualitative part and extends the analysis by relating these characteristics to the efficiency measure of issue processing times.

5. Quantitative Study

The objective of our quantitative study is to empirically check an association between issue tracking efficiency and collaboration network structure. We first introduce our variables and case-study specific measures and present and discuss the regression results afterwards.

5.1. Variables and Measures

5.1.1. Dependent Variable: Processing Time. The dependent variable of empirical studies in related work is most often some measure of corporate performance. Studies relying on sociometric data acquisition strategies in form of surveys among employees often use a manual performance rating (e.g. [4], [25]). We abstract from the people centric view of the concept of performance and are interested in process efficiency. In the context of a process-oriented organization, the people and group centric performance evaluation of social sciences does not hold anymore. We are not interested in identifying individuals or fix groups with specific performance attributes but in processes that can be seen as best practice and others that require reengineering. Our definition of economic outcome is therefore process efficiency and not employee performance.

Additional arguments for a process iteration length measure come from the specifics of the software industry. It is coined by rapid technological change and short product life cycles. In order to cope with this ever changing environment, software development processes must be executed quickly and must be as free of waste as possible. In quality assurance as a sub-process of software development, timely process execution is even more crucial. The longer unfixed bugs remain in the software artifact, the more problems are experienced by different stakeholders of the project. In a setup where known problems take long to be resolved, the product quality suffers, other groups might be hindered in their work, and the (internal) customer satisfaction decreases. We therefore use the processing length of an issue as the dependent variable in this study being in line with related work from the OSS domain (e.g. [9]).

Since the distribution of the issue processing length is skewed, we apply a logarithmic conversion that yields a normal distribution in the process efficiency variable. Our dependent variable is named $LProcTime_i$ and is the logarithm of an issue i’s processing time measured in calendar days. Processing time itself is defined as the difference between the date when an issue has been solved and the date of its initial reporting.

As explained in Section 4.1, the most suitable level of analysis for our study is that of all issues of a software component. We derive the performance measure of this group of process iterations $LProcTime_G$ from the arithmetic mean of the $LProcTime_i$, measure of all issues i contained in G.

5.1.2. Independent Variable: Communication Pattern Heterogeneity. As discussed in Section 3.1, issue tracking is a collaborative process. In this process, employees do not only search for technical solutions but also look for advice from colleagues that are deemed more capable to solve the particular issue. Our research addresses this process characteristic by analyzing both communication and collaboration network structure.

In a preparatory step to the definition of our first independent variable, we investigate the advice network absorption $AdvNetwAbsorp$, which indicates whether many different employees have participated in the processing of an issue i or whether it has been processed by few experts only. In order to control size effects, we normalize it by the number of commitments to the issue, i.e. the atomic collaborative acts by employees. The normalization further makes $AdvNetwAbsorp$ robust against the unmeasured technical complexity of
an issue—more difficult technical problems require more interventions and more time for solving than simple ones. AdvNetwAbsorp, takes values between 0 (only one soliloquizing employee) and 1 (multiple employees each contributing to the issue exactly once). In the example of Figure 1(a), the 8 commitments (completion and confirmation are counted together) and 4 distinct contributors yield an AdvNetwAbsorp of 0.5.

With the help of the advice network absorption variable, we are able to define our measure for communication. Due to conducting process group based research, we are interested in whether a group communicates homogeneously or heterogeneously over all its process iterations. Balanced communication patterns have been found to correlate with performance (e.g. [15]) and can be seen as an indicator of well established processes.

We operationalize this variable on the basis of the AdvNetwAbsorp definition given above. When analyzing the communication pattern heterogeneity of all process iterations for a group, we are interested in the variance of the advice network absorption in the different process iterations. The variable ComPatHeterog is therefore operationalized as the standard deviation of AdvNetwAbsorp for all issues i of G. Low values of ComPatHeterog indicate homogeneously communicating groups whereas high values mean that different issues of a group are solved by means of varying advice network absorption levels.

5.1.3. Independent Variable: Social Network Centrality. An additional perspective on the collaborative process of issue tracking is social network analysis. While there are many different measures available in SNA, especially centrality (cf. [32]) has been found to correlate with performance measures in related OSS case studies (e.g. [33]).

Several different definitions of this metric have emerged [19]. Degree centrality represents the number of links of a node divided by the number of nodes minus one. It is very straightforward in its calculation and expresses in a simple way how central a node’s position in a network is. Betweenness centrality is a more complex measure that represents the average frequency with which a node lays on the shortest path between two arbitrary other nodes of the network. It is a good measure of the ability to absorb information flowing through a network and of being important for the cohesion of a network. Closeness centrality is defined as the average distance of a node to all other nodes in the network. It expresses the independence of a node from others in the network due to reaching alternatives quickly (c.f. [32]).

In literature, it is often stated that the principles of social network analysis can especially be observed when being applied on large networks (e.g. [22]). We investigate this phenomenon by deriving two sets of SNA variables. Those variables with the ‘local’ attribute in their name are defined on process group level (illustrated as (b) in Figure 1). The social network matrix SNLocalG of a group G is two-dimensional and its size matches the number of distinct employees active in any of G’s issues. The fields in this matrix are 1 when the two corresponding employees jointly worked on at least one of the issues in G and 0 otherwise. Thereby, the diagonal is 0 (self-relations are not counted) and all values above the diagonal are 0 (undirected network).

The alternative definition of the social network studied in this paper is that of large networks (cf. [22] for the advantages of SNA in the large). The corresponding variables carry the ‘global’ attribute in their name. The definition of the underlying network SNGlobal is analogous to that of SNLocalG except that the network matrix comprises all users active in any issue and a 1 is set when two employees jointly worked on any issue (independent of the grouping according to software components) as illustrated in part (c) of Figure 1.

Provided with these two network definitions, we can define variables for social network centrality. To be more specific, on the level of ‘local’ groups, we are interested in centralization (i.e. an index of how centralized the entire network is) and for ‘global’ centrality we are interested in the average centralities of employees. We do not elaborate further on the concepts of centrality and centralization since they are not specific to this case study but well known concepts of SNA (cf. [32]). The specific type of centrality calculated is handled on the analysis and modeling level (we will test degree, betweenness, and closeness centrality separately).

Correspondingly, we measure LocalCentralityG as the centralization of SNLocalG and GlobalCentralityG as the arithmetic mean of the centralities of all employees active in any of G’s issues in SNGlobal.

5.1.4. Independent Variable: Number of Issues. Another independent variable is NumberOfIssuesG, measuring the number of issues that a component group has. The underlying idea is that large groups have advantages of economies of scale. They might benefit from learning effects and the availability of more specialized resources. Group size could therefore have an impact on performance. However, a potential downside might be an increase of overhead.
Table 1. OLS Regression Results for LProcTime

<table>
<thead>
<tr>
<th>Variable</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>-1.8**</td>
<td>-1.7**</td>
<td>-2.1**</td>
</tr>
<tr>
<td>ComPatHeterog</td>
<td>18***</td>
<td>18***</td>
<td>18***</td>
</tr>
<tr>
<td>LocalCentrality</td>
<td>1.5</td>
<td>2.2*</td>
<td>1.3</td>
</tr>
<tr>
<td>GlobalCentrality</td>
<td>-8.9</td>
<td>-12*</td>
<td>1.9</td>
</tr>
<tr>
<td>NumberOfIssues</td>
<td>-0.01*</td>
<td>-0.01**</td>
<td>-0.01*</td>
</tr>
<tr>
<td>Adjusted R-square</td>
<td>0.23</td>
<td>0.24</td>
<td>0.23</td>
</tr>
</tbody>
</table>

* p < 0.05, ** p < 0.01, *** p < 0.001, maximal VIF for centrality measures: B = 1.60, C = 1.04, D = 1.36

5.2. Analysis and Results

A correlation analysis of the variables in our model indicates some correlation among the independent variables of our model which is within the acceptable range (also checked by means of VIF). However, the three different centrality measures (betweenness (B), closeness (C), and degree (D)) are highly correlated among each other as expected. We therefore build separate regression models for each of these centrality measures.

Interestingly, local and global closeness centrality are related to performance in opposite directions. This behavior can be explained by the fact that employees that are close to other nodes globally and therefore well networked in the entire department are beneficial for fast issue processing. However, groups that have a high local closeness centrality might not distribute work well among its members. They rather rely on a key employee that is involved in a lot of the work done in this group. It might be better for these groups to rely on the strength of peripheral members that are rather networked to the outside than the inside (cf. [34]).

We construct different regression models in order to validate the effect of the independent variables on the dependent performance variable LProcTime. The regression results are shown in Table 1. The R-squared values are rather low, though sufficient for organizational and sociological studies [11], [35]. Despite 32% explained variance of Model 4 in the closeness centrality variant are not high enough for the model to comprehensively explain the LProcTime dependent variable, we do not attempt to holistically model the issue tracking process in its full technical complexity but only want to examine the association between dependent and independent variables.

Closeness seems to be the most appropriate centrality measure throughout models 4 to 6. We also observe that the fully specified Model 4 is to some degree superior to the variants with omitted local centrality (Model 5) or global centrality (Model 6). In all models, the communication pattern heterogeneity has a highly significant influence on the processing time. Homogeneous communication in a group appears to accelerate processes. In Model 4C, both centrality metrics have a significant influence on the performance measure. While local centrality is impedimental, employees’ centrality in the case study wide network is beneficial for quick process executions. Throughout all models, the size of the group (measured in handled issues) is negatively associated with LProcTime. The more process iterations a group has to execute, the faster they are.

5.3. Quantitative Findings

Our empirical study reveals several characteristics of the analyzed issue tracking process. First, heterogeneity in the communication patterns (resp. advice network absorption) yields longer processing times. A possible explanation for this behavior is the immaturity of those groups’ processes. A best practice of the amount of communication necessary to solve issues quickly has not yet been found.

We also found that centralization of the ‘local’ collaboration network slows issue cycling times down while average centrality in the ‘global’ network speeds it up. It is beneficial for a group to count on members that are connected in the case study wide collaboration network. However, the more centralized a group’s network is, the slower its issue processing. This can be explained by the introduction of a bottleneck within a group as discussed in the qualitative study of Section 4. When many issues need to pass one employee, he or she is the limiting resource and the entire group’s issue tracking velocity decreases. [20] also name increased overhead as an additional negative effect of centrality.

A last quantitative finding is that the more issues
a group processes, the faster is the average processing. This could be explained by a comparison with economies of scale. Employees active in these groups can leverage scale effects and avoid set-up costs that can occur in case of frequent process or task switches.

6. Discussion & Conclusions

In this study, we address the recent interest of the IS community in the association of social network structure and economic outcome. We conduct a case study at a large business software vendor. While few related studies originate from an industrial setup, many utilize academic data sets or open source software development repositories. We argue that agile process organizations, short product life cycles, and rapid technological change require an unobtrusive data acquisition strategy. Sociometric surveys are found to not be a suitable strategy due to the size and complexity of the network organization in large software development units. We rather utilize data readily available in issue tracking systems. They cover large parts of the software quality assurance process which is an important sub-process of software development in terms of effort and involvement of the entire development organization.

Qualitatively analyzing our data, we find that the most suitable level of analysis are collaboration networks of employees resolving issues of specific software components. These networks’ vertices are all employees that have been involved in at least one issue of the corresponding software component and two vertices are connected when the two employees have jointly worked on the resolution of a common issue of that component. We identify variance in structural properties of these networks and find that centralization slows the issue tracking process down in many cases. Sample issues and collaboration networks reveal that this is often due to bottleneck effects that are caused by the assignment of a dispatcher role to a specific employee for a specific software component. This dispatcher routes new issues to other employees before they begin to collaboratively resolve the problem.

In order to empirically validate our research question of the association of collaboration network structure and issue tracking’s process efficiency, we additionally conduct an empirical study. Our results confirm the qualitative findings by suggesting that centrality in the organization-wide networks, communication pattern homogeneity, and economy of scale effects are positively associated with issue tracking’s process efficiency. We find centralization of small groups of collaborating issue processors to be negatively associated with issue tracking performance. This finding confirms our bottom-up process understanding that central employees like issue dispatchers can increase overhead and build bottlenecks.

Since distributed teamwork is present in many software development organizations, we believe that our findings are valid in other industrial setups as well. Transferability is also supported by studies from the context of open source software development that confirm parts of our findings.

Possible limitations of our empirical study can be seen in insufficient model specification and undetected time dependencies. Despite a thorough variable definition, we are aware that the complex process of issue tracking cannot be fully grasped by this variable selection. We therefore limit our findings to answering our research question and do not intend to comprehensively model the dependent variable of process efficiency.

Future research should concentrate on the definition of further variables of the issue tracking process and experiments with more sophisticated measures of social network analysis. We also suggest comparative studies with other industrial software development units and possibly with open source projects.

References


