How to Implement Clan Control in DevOps Teams

Completed Research

Anna Wiedemann
Neu-Ulm University of Applied Sciences,
Germany
anna.wiedemann@hs-neu-ulm.de

Manuel Wiesche
Technische Universität München,
Germany
wiesche@in.tum.de

Abstract

Clan control is an important factor for project success. A clan is a group with strong social capital. Hence, the implementation of social capital is a precondition for clan control and project success. However, for managers it is still a challenge to build and motivate clan control within IT projects, because people have different skills and backgrounds. Especially in rapid application development projects when new project methods or approaches like agile software development or DevOps are applied. Thus, we build up on existing research and explore how clan control can be built and leveraged within cross-functional DevOps teams. With the help of expert interviews with team leads and managers of different DevOps teams, we derived structural, cognitive and relational ties of social capital. Additionally, we present leverage factors from our data for social capital to facilitate clan control. Finally, we give implications how these factors can be implemented.

Keywords

DevOps, IT project management, Clan control, Social capital

Introduction

More and more organizations organize work in team based structures (Kirsch et al. 2010). Reasons therefore are, that work gets increasingly complex, knowledge intensive and depends on non-routines. With the help of teams, organizations want to react to changing situations (Goh et al. 2013). Teams compose of individual people with different knowledge as well as skills and can act in a collaborative manner to realize a specific aim (Kirsch et al. 2010; Klendauer et al. 2012; Towry 2003; Wiesche and Krcmar 2014). IT project success is still an aim that is difficult to achieve.

One possibility to achieve collaboration is control, it “is defined as any attempt to align individual behaviors with organizational objectives” (Wiener et al. 2016, p. 742). To achieve a desired behavior, formal as well as informal control can be used (Heumann et al. 2015; Kirsch 2004; Ouchi 1978). In context of teams, it is difficult to rely only on formal controls because of individual behaviors and the complexity to measure personal contributions to team success (Kirsch et al. 2010; Schermann et al. 2012; Towry 2003). Information systems (IS) project control is essential to drive or adjust project stakeholders’ behavior to motivate them to achieve project aims (Kirsch et al. 2010; Wiener et al. 2016). The literature highlighted that there is a relationship between social capital and informal clan control (Chua et al. 2012; Kirsch et al. 2010; Liu et al. 2015; Wiener et al. 2016).

Clan control is defined as a type of informal control. It appears when the team members behavior is motivated by shared norms, values and strong affiliation feelings to the group with a common goal (Kirsch et al. 2010). Clan control is a phenomenon that occurs on team level. Teams’ social factors play an important role. Moreover, the literature highlighted that social factors are an essential prerequisite of clan control and next to team members, managers play an important role to generate clan control (Kirsch et al. 2010; Towry 2003). Managers are often part of the team when they work within or very close with the team (Kirsch et al. 2002; Liu et al. 2015).
In the recent years, the use of agile methods in IT projects gained wide acceptance. It has become a major driver for performance within a lot of IT functions (Cram and Newell 2016; West et al. 2010). Software development projects remain a major concern for IT managers. Today, companies like Kaiser Permanente apply an approach called DevOps that goes beyond agile (Ross et al. 2016), it is compound of “development” and “operations”. This approach helps IT departments to transform to service-centric IT operating models. With the help of DevOps, IT departments are enabled to reduce their software cycle times and bring new features into production in very short time (Ross et al. 2016; Sebastian et al. 2017).

The literature highlight that implementing DevOps capabilities will become a competitive requirement for incumbent companies (Sebastian et al. 2017). Managers are challenged to bring team members closer together for achieving project success (Chua et al. 2012; Majchrzak et al. 2005). Prior research depicts that social capital could be helpful for enhancing team work (Liu et al. 2015). Social capital is defined as “the sum of the actual and potential resources embedded within, available through, and derived from the network of relationships possessed by an individual or social unit. Social capital thus comprises both the network and the assets that may be mobilized through that network” (Nahapiet and Ghoshal 1998, p. 243). Since project members often have different backgrounds from several professions that need to achieve culture of collaboration (Chua et al. 2012; Wiesche and Krcmar 2014). Within DevOps teams barriers between development and operations people could appear. There is a need to solved these barriers to gain effective collaboration (Fitzgerald and Stol 2017). Often complex tasks need to be addressed by the team, therefore strong collaboration with team members and shared values, understanding as well as social cohesions are necessary (Liu et al. 2015). Hence, we put forth the following research questions: How can managers build and leverage social capital within DevOps teams to achieve high clan control?

We argue that managers can promote clan control within team settings for example team-based clan control through the usage of social capital and the implementation of leverage factors. We derived by these factors with the help of a qualitative investigation IT managers that are responsible for DevOps teams. We investigate how social capital influence the collaboration within DevOps teams. Furthermore, we provide three factors that influence the relationship between social capital and clan control.

Related Literature

DevOps Teams

Since the presentation of the agile manifesto in 2001, agile software development methods gained high popularity (Tripp et al. 2016; West et al. 2010). A prerequisite of agile development teams is the ability to react to new and unforeseen situations, for example new customer demands, with the help of a team (Wiedemann and Weeger 2017). The aim is to handle the complexity of fast changing environments and to achieve better software quality as well as higher customer satisfaction. Thus, more and more IT functions are implementing product-oriented agile IT teams (Burke et al. 2006; Lassak et al. 2017; Pflügler et al. 2018; Przybilla et al. 2018).

The literature shows that research on agile methods mainly focuses on development activities (Goh et al. 2013; Tripp et al. 2016). IT functions have to enable a tighter collaboration between the different units of development and operations of an IT function to ensure that errors are fast fixed. Hence, the quality and resilience of the software is enhanced. The literature presents that IT functions’ activities of development and operations have to be continuous (Fitzgerald and Stol 2017). DevOps helps to combine these approaches. For fast providing of new software features and reacting on problems, IT departments should implement cross-functional teams rather than separated silo IT departments (Fitzgerald and Stol 2014).

Within many IT organization customers gain deployments of new software code very seldom (Lwakatare et al. 2016). One reason is the poor exchange in form of communication and collaboration between the two department of software development and operations (Fitzgerald and Stol 2017). To address this issues, DevOps provides solutions to avoid interruptions between different stages of the software delivery process (Fitzgerald and Stol 2014). The software development life-cycle involves planning, building and, running processes. DevOps helps companies to implement speed and flexibility to deliver rapid development and implement digital innovations when they are needed (Ross et al. 2016). Hence, possible risks through software releases can be reduced, and feedback on a new software deployment can be faster.
Building and leveraging clan control in DevOps teams

Social capital and Clan Control

Social capital theory is important for collaboration and addresses the social and cultural preconditions (Riemer and Klein 2008). The literature presents that managers can support and influence a project by social capital. Two forms of control exist, namely formal and informal (Kirsch et al. 2010; Ouchi 1979). Formal control focuses on hierarchical power and authority that influence team members to act in a specific way. Informal control relies on self-control through individuals characteristics and/or social relationships (clan control) (Kirsch et al. 2010; Liu et al. 2015). Prior research presents that there are relationships between social capital and clan control (Chua et al. 2012; Kirsch et al. 2010). A clan is defined as a homogenous group with individuals that share common beliefs, values and norms (Liu et al. 2015; Ouchi 1979).

Social capital can be measured with the help of three dimensions namely: structural, cognitive, and relational dimensions. These dimensions can be implemented as interrelated ties between team members (Nahapiet and Ghoshal 1998; Wagner et al. 2014). The structural dimension of social capital comprises the contact between the individuals of the network (Nahapiet and Ghoshal 1998). Structural ties include the settings and forms of communication meetings of the project members which are typically described by frequency, centrality, stability, and density. Communication exchange can happen physically by colocation, or virtually by emails (Liu et al. 2015; Wagner et al. 2014). The cognitive dimension is defined as the assets that deliver shared understanding and importance among the stakeholders (Nahapiet and Ghoshal 1998). This dimension includes common language, shared codes, narratives and perspectives as well as the project stakeholder’s interpretation of reality (Wagner et al. 2014). The relational dimension views on the specific relationships that the people of the relationship have (Nahapiet and Ghoshal 1998) and includes trust and respect which is a precondition for knowledge sharing. Furthermore, the stakeholders see each other as partners and consult each other for better working together (Wagner et al. 2014).

Social capital is related with networks and relationships between individuals which enhance collaboration. Prior research treats groups with strong social capital and clans as interchangeable (Chua et al. 2012). Managers can be a part of clans if they work close together with the members (Kirsch et al. 2002). Clan control can be established by the building of social capital. Managers should guide social capital in the clan to reinforce shared values, beliefs and norms that are helpful for project success (Chua et al. 2012). Enacting clan control is bilateral, by building and leveraging the clan. In the first step social capital is developed by structural, cognitive, and relational ties and in the second step, social capital can be leveraged for a specific outcome (Chua et al. 2012; Liu et al. 2015). Within this paper, we derived factors for social capital, afterwards we present how these factors can be leveraged. This research focuses on clan control, because prior research depicts managers could be team members too. We are interested in building relationships between team and manager (Liu et al. 2015). Hence, we examine the development of for example shared norms, values and do not focus on self-control characteristics like self-monitoring of behavior (Wiener et al. 2016).

Research Methodology

To answer our research question, we decide to conduct an exploratory research. By using a qualitative research approach, our research can be characterized as interpretive. We want to understand how social capital influences DevOps teams and how they are implemented within existing IT function. Hence, we conduct a row of expert interviews with managers and team leads of DevOps teams. Due to the fact that DevOps is a very unexplored research topic, it calls for case study design (Yin 2009). The case study approach is defined as "an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context" (Yin 2009, p. 18). Case studies are a suitable approach in rare areas in which researcher as well as practitioners search for new insights (Cao et al. 2013; Eisenhardt 1989). Thus, new findings and a deeper understanding of the topic can be achieved. DevOps settings have to be investigated in their daily environment to gain insights about that approach and how the teams are received (Lwakatare et al. 2016). To summarize, the aim of the DevOps concept is to foster collaboration, automation, virtualization as well as implementing tools for bringing activities of software development and operation closer together (Humble and Molesky 2011; Lwakatare et al. 2016).
working with that concept. Hence, qualitative methods are appropriate if the existing body of knowledge lacks of information and questions are open, which need further investigations.

**Site Selection**

For the investigation, the focus laid on DevOps teams. Therefore, we identified contact persons, which are engage to the DevOps concept. The focus laid on people, who have experience with DevOps and management position within a IT organization. For the expert interviews, over 50 companies from different industries were contacted via e-mail and telephone, the premise for study participation was that they have already implemented a DevOps team. Our aim was to find cases that have an interesting setting or are easy to replicate to achieve similarities and variation about our findings (Eisenhardt 1989). In each company, minimum one manager (e.g. team lead) was interviewed regarding their control mechanisms of DevOps teams. The aim was to gain the leadership view on how social capital influence the social relationship with DevOps teams. In summary, we talked to ten managers from seven companies with help of eight interviews (interviews in company 1 and 2 were conducted with both managers at the same time).

<table>
<thead>
<tr>
<th>ID</th>
<th>Role</th>
<th>Team</th>
<th>Work Experience</th>
<th>Industry</th>
<th>DevOps since</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Manager</td>
<td>Team 1</td>
<td>10 -15 years</td>
<td>Media</td>
<td>six month</td>
</tr>
<tr>
<td>I2</td>
<td>Manager</td>
<td>Team 1</td>
<td>10-15 years</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3</td>
<td>CTO</td>
<td>Team 2</td>
<td>5-10 years</td>
<td>Furniture</td>
<td>two years</td>
</tr>
<tr>
<td>I4</td>
<td>Manager</td>
<td>Team 3</td>
<td>5-10 years</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5</td>
<td>Manager</td>
<td>Team 3</td>
<td>15-20 years</td>
<td>Fashion</td>
<td>three years</td>
</tr>
<tr>
<td>I6</td>
<td>Manager</td>
<td>Team 4</td>
<td>10-15 years</td>
<td>Service</td>
<td>five years</td>
</tr>
<tr>
<td>I7</td>
<td>Manager</td>
<td>Team 5</td>
<td>10-15 years</td>
<td>Energy</td>
<td>&gt; five years</td>
</tr>
<tr>
<td>I8</td>
<td>Manager</td>
<td>Team 6</td>
<td>10-15 years</td>
<td>Health Care</td>
<td>three years</td>
</tr>
<tr>
<td>I9</td>
<td>Manager</td>
<td>Team 7</td>
<td>15-20 years</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I10</td>
<td>CTO/Manager</td>
<td>Team 7</td>
<td>10-15 years</td>
<td>Media</td>
<td>two to three years</td>
</tr>
</tbody>
</table>

**Table 1. Characteristics of the Interview Partners**

Table 1 depicts that we interviewed managers from different industries. The interviewees had different positions for example CTO or manager. Managers are members of the team and/or responsible for one or more teams. The cases have experience with the DevOps approach between six month and five years.

**Data Collection and Analysis**

The data collection phase took place from October 2016 through November 2017. After identifying possible interviewees, semi-structured interviews were conducted with the participants. Each interview had a duration about 45-75 minutes and was carried out primarily through face-to-face meetings or by telephone. In exploratory research personal interviews are recommended because they allow comprehensive discussions. The interviews were held in German or English language. German statements were translated into English for further analysis. The questions were mainly open-ended, that the interviewee had the possibility to explore their experiences and views (Yin 2009). The questionnaire comprises questions regarding general background of the organization, the use of DevOps principles, and the implementation and enhancement of social capital (e.g. how do the stakeholders communicate?). The interviews delivered insights into the working style of the teams and informal control mechanisms of the DevOps team. Every interview was recorded.

With the help of the expert interviews with the IT managers, we were able to derive general constructs for social capital as well as factors that leverage clan control. We used coding processes to investigate the relationships between managers and DevOps teams. Coding is very helpful to identify categories and corresponding sub-categories and for examining relationships (Runyan et al. 2007; Sarker et al. 2013). The interview data was coded using the software NVivo 10. We started the coding process following the guidelines of Miles and Huberman (1994). Hence, we started with an open coding process. During the
Building and leveraging clan control in DevOps teams

coding, the research team took notes to justify the coding section. Afterwards, the results were analyzed regarding the presented ties of social capital. Then the research team compared their findings for each dimension to identify commonalities, relationships and patterns. The focus laid on the constructs which we identified from literature and the new capabilities that emerged during the data analysis. Furthermore, we analyzed the findings regarding relationships to identify the degree leverage factors within the teams.

Findings

We analyzed the cases to identify which forms of social capital are established to enhance relationships through the managers and DevOps teams. We focus on social capital that is built by the managers in each DevOps team to generate ties (Liu et al. 2015; Nahapiet and Ghoshal 1998). Table 2 provides insights how social capital can be built by managers and are helpful to build effective clan control (Chua et al. 2012).

<table>
<thead>
<tr>
<th>Construct</th>
<th>Constructs</th>
<th>Example Quotes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural ties</td>
<td>Enable physical and virtual meetings that support the group.</td>
<td>“We have eight people in India and two in Germany, [...] they work very close together on a cross-border and daily basis. At the moment we have an overlap of four hours that is very good.”</td>
</tr>
<tr>
<td></td>
<td>Integrate autonomous teams with product responsibility.</td>
<td>“We are in the process of integrating completely self-organized teams, which means that they are organizationally and technically independent.”</td>
</tr>
<tr>
<td></td>
<td>Implement different forms of meeting structure (team internal and external).</td>
<td>“Lectures outside the box so that you look at new tools, new frameworks, new developments.”</td>
</tr>
<tr>
<td>Cognitive ties</td>
<td>Spread knowledge within the team and company.</td>
<td>“We do not have pockets of knowledge. Of course we try to avoid that very much. So it is the worst case, if something like that happens.”</td>
</tr>
<tr>
<td></td>
<td>Establish cross-functional teams with shared knowledge.</td>
<td>“We do not always have everything to do top down by default we can look how other teams do that.”</td>
</tr>
<tr>
<td>Relational ties</td>
<td>Generate awareness of product ownership.</td>
<td>“That’s the part that I’m proud of - we noticed that the teams are going to grow beyond themselves and also bear extra effort to deliver the service in a high quality.”</td>
</tr>
<tr>
<td></td>
<td>Enable freedom within the team.</td>
<td>“It is nice to see how this freedom develops the team positively.”</td>
</tr>
<tr>
<td></td>
<td>Give responsibility and trust into the team.</td>
<td>“Management positions will break away and there is a social responsibility towards managers [...] that are 15 years or longer within the company.”</td>
</tr>
</tbody>
</table>

Table 2. Social Capital used in DevOps Teams

Table 2 presents activities that helps to build a clan but our findings present that distinct actions of managers are necessary to motivate the team after building social ties. This is in line with the findings of Chua et al. (2012) there are two ways of managing a clan proactively. The first one is to reinforce shared norms, values and beliefs that are important for projects and the second one is to constrain beliefs, norms and values that endanger project success. Table 3 presents insights of the purpose and necessary steps that were used by the managers to motivate the team to achieve high clan control. The order presents the frequency and importance of the constructs, for example call for operations responsibility was a major concern of our informants.

<table>
<thead>
<tr>
<th>Leverage factors</th>
<th>Purpose for managers</th>
<th>Necessary Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call for operations responsibility</td>
<td>Team members need to overtake responsibility for operations.</td>
<td>Build the awareness for the complete software delivery life-cycle. Team members are equally responsible for development and operations tasks. But the fear of being overstrained should be avoided. If the system is not working, the</td>
</tr>
</tbody>
</table>
Despite being a clan member, for integrating clan control in DevOps settings a suitable management style is necessary. Our findings give insights that the manager role is changing in this flat hierarchies. That means that the manager is seen as leadership person as well as a team member. Team lead I6 mentioned that “there are very few situations where I really have to lead and control the team” only in situations of for example major incidents the team leads “phone is ringing first.” Additionally, one manager highlighted that “an authoritarian leadership style in the development team is absolutely not working. Laisser-faire always has a bit of an approach that it is not effective. What actually comes out of my experience is an intrinsic motivation” I7. The literature presents that control is necessary over several hierarchical levels and that cross-level difference exists in the control styles. Further, there exist different forms of control styles, coercive or enabling (Heumann et al. 2015). Hence, we studied different control mechanisms that are helpful to find out how the control style could be influenced. We highlighted that both forms of control styles are necessary. For example our findings and leverage factors present that for managing problems, clear guidelines need to be defined, and for integrating culture alignment, enabling styles are suggested. Hence, we summarize that a suitable control style is essential in DevOps teams that fosters intrinsic motivation, create awareness, willingness and attitude for the DevOps approach. We

| Integrate a culture of feedback and learning | Team members must work within feedback culture and enable continuous learning. | Foster the willingness to give constructive feedback and move from a finger-pointing culture to a feedback and learning culture. |
| Establish receptiveness for fast changes | Team members need to react as soon as possible to changing demands or failures. | Enhance the attitude for continuous delivery in fast changing environment. Team members have to learn new actions from development and operations as well as stay up to date to handle new complex changes. |

<table>
<thead>
<tr>
<th><strong>Table 3. Leverage Factors, Purpose and Necessary Steps for Managers</strong></th>
</tr>
</thead>
</table>

Table 3 depicts the norms and values that could be fostered by managers to leverage clan control. Despite a high level of social capital, these factors could help DevOps teams to achieve a high level of clan control.

### Discussion

Our study investigates the implementation of social capital within cross-functional teams. We examined how managers can foster social capital within DevOps teams and what are major challenges for achieving clan control. Now, we want to discuss the major challenges for building and leveraging social capital. Our findings suggest that high clan control is an important factor, because in DevOps settings, team leads move toward a team member position. Nevertheless, our results present that there is still a need for management and guidance of the DevOps team. In the following sections we explain how these relationships can be established.

Our findings indicate that managers have to make great efforts to implement new activities within the teams. For example, integrating operations activities into a development team is a great challenge. Prior research depicts that software operations and development differs in many ways (Edberg et al. 2012). For example developers wanted to implement fast new software features, but operations people want less changes in the system to guarantee stability (Edberg et al. 2012; Shaft and Vessey 2006). Hence, development and operations are usually controlled with the help of different modes. These tensions could complicate the building of a clan within projects. We present constructs of social capital and clan control to explain the approximation of different backgrounds that could be achieved through DevOps.

Furthermore, our study confirmed existing research that highlights that managers can be part of the clan (Liu et al. 2015). If the managers have a high business orientation, the product owner part of an DevOps team might be suitable position for them. Product owners are usually responsible for changes, refine and prioritized the product backlog, which is a list with new tasks that should be built for the system (Schlauderer et al. 2015). I10 demonstrate that he has overtaken the product owner role additionally to his CTO position and the “team together with the product owner overtake all necessary roles” for managing the service.

Despite being a clan member, for integrating clan control in DevOps settings a suitable management style is necessary. Our findings give insights that the manager role is changing in this flat hierarchies. That means that the manager is seen as leadership person as well as a team member. Team lead I6 mentioned that “there are very few situations where I really have to lead and control the team” only in situations of for example major incidents the team leads “phone is ringing first.” Additionally, one manager highlighted that “an authoritarian leadership style in the development team is absolutely not working. Laisser-faire always has a bit of an approach that it is not effective. What actually comes out of my experience is an intrinsic motivation” I7. The literature presents that control is necessary over several hierarchical levels and that cross-level difference exists in the control styles. Further, there exist different forms of control styles, coercive or enabling (Heumann et al. 2015). Hence, we studied different control mechanisms that are helpful to find out how the control style could be influenced. We highlighted that both forms of control styles are necessary. For example our findings and leverage factors present that for managing problems, clear guidelines need to be defined, and for integrating culture alignment, enabling styles are suggested. Hence, we summarize that a suitable control style is essential in DevOps teams that fosters intrinsic motivation, create awareness, willingness and attitude for the DevOps approach. We
recommend further research in the area of which control styles are suitable for different control situations.

The literature gives insights in how to include social capital into virtual organizations to gain credibility (Riemer and Klein 2008). Our findings indicate that there is a necessity of structural ties too. Some managers highlighted that they have to organize virtual meetings because they work in offshore (team 6) and nearshore (team 1) team settings. To foster social capital, the interview partners mentioned that they organize regular personal meetings with the near- and offshoring colleagues to share knowledge, communicate and “there is almost every week either someone from there here or someone from here there. I was there last week and that’s very, very important. Especially with new concepts it is really much more effective to spend a week together” I6. Hence, we summarize that virtual meetings are helpful for the management of existing and stable projects, but if new concepts like DevOps are used it is helpful to meet each team members personally.

Within the area of cognitive ties our findings present that a common and shared understanding as well as knowledge about the tasks of the software delivery life-cycle are indispensable. CTO I3 mentioned that the “team is responsible from planning, operations, development, tests, and later for monitoring. The tasks are completely shared in the team.” A high degree of autonomy regarding decision making is typical in agile development projects (Cao et al. 2009). But within DevOps the team has to make decisions about their service that could have influence to the rest of the company. Hence the team must be able to communicate with other teams and managers if decisions “concerns the system landscape” I3.

Regarding relational ties, our findings confirm insights from prior research that trust is a major factor for successful team collaboration (Liu et al. 2015). Managers have to trust the team if problems appear “the team tries to solve it, [if they cannot] an escalation processes will be started […] this could not be achieved with a command and control leadership style” I4. The “ownership for the service” I4 is changing because the complete team is responsible for the service and hence, a feeling of responsibility within in the team can be established.

For a high motivated and aligned team leverage factors are necessary (Chua et al. 2012). In our research we identified three leverage factors that are used by managers to achieve high level clan control: call for operations responsibility; integrate a culture of feedback and learning; establish receptiveness for fast changes. These factors should be considered by the managers when a DevOps team is already established. Managers should have actions implemented to build awareness “the learning curve should be not too high and [new tasks] should be slowly acquired” I6. Implementing DevOps approaches need a balance for the call for new responsibility and creating awareness for this necessity. The call for operations is an important leverage factor for managers. DevOps describes the development and operations tasks of the software delivery lifecycle, the team members have to overtake new responsibility. It mentioned that great efforts are necessary to “claim for the operations responsibility” of the team members, because a lot of team members are not willing to overtake this new responsibility.

Prior research shows that tight project schedules are often insufficient for building a clan (Chua et al. 2012). However, our findings present that DevOps teams are often organized in so-called product teams rather than in project teams. That means that they have no clear start or end date and the team members have to work there for an undefined time. Hence, we derive that clan control can be fostered in DevOps settings, because the ongoing team structure presents freedom and time for developing a clan. Additionally, in DevOps team settings it is important to deliver fast feedback to the team because there is usually a high degree of automation (Fitzgerald and Stol 2017). In project 1 “these feedback loops were introduced, because we really put the operations pain into the development team and that first hurts and then it gets better” I2. The team members need to know if something went wrong. Within DevOps settings, the team members should “work in an open feedback culture and […] always try to make decisions by consensus” I10 and work on a common solution. To summarize, the integration of a culture of learning and feedback seemed to be a key factor for leveraging clan control. Furthermore, the receptiveness for fast changes is necessary since team members need to be motivated to work in a DevOps team. Team members need the attitude to steadily work on new solutions and provide it to production in short cycle times (Fitzgerald and Stol 2017). “The most important thing is that people fit [in our team] “I10 from a personal perspective. Therefore, different measures are possible for example the integration of “an agile coach for the organization” I3, that helps teams to learn the DevOps principles and distribute knowledge within the company.
Figure 1 presents an overview of the interplay between social capital and leverage factors for clan control.

![Figure 1. Clan Control as Process of Social Capital and Leverage Factors](image)

**Implications for Research and Practice**

Our findings present insights into how managers can influence the build and motivation of clan control within DevOps teams. The present study enforces the necessity of informal control mechanisms in IT teams (Chua et al. 2012; Liu et al. 2015) and depict how social capabilities can be established and leveraged within DevOps teams. We contribute to existing research and enhanced prior findings of Chua et al. (2012) and Liu et al. (2015). We follow their call for further research in different project characteristics of different organizations and present an investigation of DevOps teams. Management of DevOps teams have an influence to motivate clan control as a controller to build structural, cognitive and relational dimensions of social capital. We deliver observations of social capital that could be facilitated by the managers within DevOps team settings and present a basis for further research. We enhance existing research on clan control in cross-functional IT teams and present DevOps specific social capital and three leverage factors that help to achieve clan control within these teams. Through the present paper an instigation of managing teams regarding their activities of the software delivery lifecycle with focus on software development as well as operations tasks is provided.

For practice we present guidelines for managing DevOps teams from an informal control perspective. We describe the changing role of a management positions to a more coaching and team member oriented role. Additionally, we identified challenges from our data and subsequently actions for managers how they can handle and motivate clan control. We derived three factors that leverage clan control if they are combined with the ties of social capital.

**Limitations**

The study provides insights into the clan control of DevOps teams, but has some limitations that need to be considered when interpreting the results. The generalizability of the findings is limited, because we only investigated one team and one or two manager perspectives in seven companies, this means that we only examined one DevOps team per organization. In addition, all the organizations are located in Germany. This entails that the applications of the results are limited. Further research could repeat the study in different countries, to examine more teams and different settings per company. Furthermore, a quantitative research could achieve validation for our findings.

**Conclusion**

In this paper we demonstrate how managers can support collaboration within IT teams and build social relationships. With the help of DevOps teams, we examined the influence of social capital for collaboration. Managers should foster structural, cognitive and relational ties through concrete actions. We derived constructs of our data from expert interviews and depict the importance of social capital within project teams. For every tie of social capital, we delivered several constructs that provide insights on how social capital can be built with help of managers to support an IT team. Furthermore, we derived leverage factors for achieving a higher clan control if they are combined with social capital of the team. The three factors are: call for operations responsibility, integrate a culture of feedback and learning as well as establish receptiveness for fast changes. If managers could implement these factors and achieve awareness, willingness, and attitude for the DevOps approach, clan control could be leveraged.
Additionally, we described the changing role of managers within DevOps settings. Managers have to lead the teams on the one hand and act as team members on the other hand. These findings present possibilities for further research for example key characteristics of the leadership style for cross-functional DevOps.

**Acknowledgment**

We thank the German Federal Ministry of Education and Research (BMBF) for funding this research (grant code 03FH005PX4). The responsibility for the content lies with the authors.

**REFERENCES**


Building and leveraging clan control in DevOps teams


