MUDSLINGING AND MANNERS

ANNA FILIPPOVA

NATIONAL UNIVERSITY OF SINGAPORE

2016

MUDSLINGING AND MANNERS

A MULTI-METHOD EXAMINATION OF CONFLICT IN FREE AND OPEN SOURCE SOFTWARE DEVELOPMENT

ANNA FILIPPOVA

B.A. (Hons), Monash University, 2008

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

COMMUNICATIONS AND NEW MEDIA

NATIONAL UNIVERSITY OF SINGAPORE

2016

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Anna Filippova
April 22, 2016

Name	:	Anna Filippova
Degree	:	Doctor of Philosophy
Supervisor(s)	:	Dr. Hichang Cho
Department	:	Communications and New Media
Thesis Title	:	Mudslinging and Manners A multi-method examination of conflict in Free and Open Source Software Development

Acknowledgment

Foremost, I would like to express my sincere gratitude to my advisor Dr. Hichang Cho for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Jude Yew and Dr. Michael Stefanone, for their encouragement, insightful comments, and hard questions.

My sincere thanks also goes to the Singapore tech community for offering me their hospitality, friendship and support.

Last but not the least, I would like to thank my family: both the one I started with four years ago, and the one I found along the way. You are my biggest cheerleader.

April 22, 2016

Contents

Abstract

This dissertation presents evidence to expand theory on conflict in ongoing distributed virtual work groups, such as Free and Open Source Software (FOSS) communities, through the inclusion of normative conflict. Specifically, the work outlines how FOSS teams experience disagreements not only arising out of the direct action of working on a software project, but also concerning the norms of the group, such as ideology, mission and values. Conflict as a team process is thus more than a task byproduct; in long-term oriented self-organizing teams, it is a self-reflexive process that can stimulate community evolution, and in some cases, dissolution.

Normative conflict is not only a novel addition to conflict theory, but also a factor with significant impact on the health of open collaborative communities. Empirical findings suggest normative conflict to be the strongest overall predictor of developer retention, having both a direct negative effect, as well as through exacerbating other types of work-driven conflict.

The dissertation traces the discovery (through interviews and community engagement), operationalization (through development and refinement of a normative conflict scale) and generalization (following a wide survey) of normative conflict as part of an updated conflict framework for ongoing distributed virtual work groups.

The dissertation thus presents a theoretical, methodological and practical contribution to understanding conflict in virtual communities and organizations. Theoretically, the work elucidates and provides support for the expansion of our existing understanding of conflict in distributed teams. Methodologically, the work provides a validated set of measurement tools for the updated conflict framework. Finally, the dissertation examines various team structural factors that may stimulate or reduce conflict occurrence, offering practical advice for community building and maintenance.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

1.1.1 Problem

The virtual workforce is continuing to grow. The number of employees telecommuting has virtually doubled between 2012 and 2014, amounting to 6.5 million people in the US, or 4.5% of the total workforce (*American Community Survey*, 2014). At the same time, virtual work is evolving, and so must our understanding of the foundational processes that underpin on-line collaboration. Distributed teams are increasingly shifting away from fixed membership towards dynamic composition (Tannenbaum, Mathieu, Salas, & Cohen, 2012; Wageman, Gardner, & Mortensen, 2012). Team boundaries are becoming more fluid, while members are more likely to be assigned to multiple teams (Tannenbaum et al., 2012). Crucially, virtual teams are becoming increasingly less likely to experience life cycles with distinct start and end points and more likely to expect working together across multiple tasks and projects (Saunders & Ahuja, 2006). Because virtual team interactions are a product of both the technical foundations underpinning their communication, as well as the social structures that emerge (DeSanctis & Poole, 1994), the above structural changes are likely to introduce substantial shifts in the social processes of virtual teams.

Conflict, or a manifested perceived incompatibility in ideas, behavior or

belief between two or more parties, is one such critical social process (Fink, 1968; Putnam & Poole, 1987). Conflict is an important social process to examine because it influences subsequent group behavior and outcomes in both negative and positive ways (Amason, 1996; Pondy, 1967), as well as being instrumental in the development of the group as a whole (Arrow, Poole, Henry, Wheelan, & Moreland, 2004; Gersick, 1988). Specifically conflict can both stimulate the production of new ideas as well as lead to feelings of frustration, reduced productivity and higher turnover. However, while conflict has received significant attention from literature on temporary, or short-term, virtual teams in traditional organizational settings (Furumo, 2008; Kankanhalli, Tan, & Wei, 2007; Montoya-Weiss, Massey, & Song, 2001), relatively less attention has been given to conflict in virtual groups with voluntary and dynamic membership composition and long term goals that can progress for several years (Gilson, Maynard, Jones Young, Vartiainen, & Hakonen, 2014; Martins & Schilpzand, 2011). Given the potential impact of conflict on turnover, it is particularly important to understand this process in groups that depend on voluntary contributions.

1.1.2 Context

Free and Open Source Software (FOSS) development represents a unique opportunity to examine conflict in dynamic, ongoing and distributed groups in a natural setting. FOSS teams are groups of individuals who work together to produce software, enabled by computer-mediated communication and permissive licenses that explicitly allow modification and redistribution. FOSS teams are highly virtual and prefer to make key coordination decisions online as far as possible (Crowston, Howison, Masango, & Eseryel, 2007). Due to the largely voluntary nature of participation, FOSS teams are often highly distributed and have fluid team boundaries (R. A. Ghosh, Glott, Krieger, & Robles, 2002). Furthermore, FOSS teams rarely have fixed end points, with notable projects continuing for several years, and even decades (Crowston, Wei, Howison, &

Wiggins, 2012). FOSS teams therefore present a useful context for studying processes in real world dynamic and ongoing virtual groups.

FOSS projects have an every day impact on the average Internet user, therefore it is critical to study conflict and its impact on participation in this setting. Technologies such as Linux, Apache, Sendmail and MySQL underpin a majority of websites (Hendrickson, Magoulas, & O'Reilly, 2012). For example, as of August 2015, the open source Apache web server maintains its decade-old lead as the most popular web server, powering 37.5% of all websites, and over half of the top million busiest sites (Netcraft, 2015). Economically, the use of Free and Open Source Software can amount to a 2:1 cost advantage relative to proprietary solutions, according to a report based on web-hosting data (Hendrickson et al., 2012). The two most popular web browsers, Google Chrome and Mozilla Firefox are both open source projects, and together amount to 85.2% of the market share as of August 2015 (W3Schools, n.d.). On the desktop front, many government organizations around the world, such as Singapore's Ministry of Defense, have transitioned to FOSS alternatives to Microsoft's Office suite to take advantage of open document standards and document longevity (Marson, 2004). Five out of the top 10 most popular programming languages are open source projects because they have open source interpreters/compilers and specifications (Cass, 2015). There are also countless libraries and applications that millions of users depend on. Given that conflict is an important process that can influence the sustained development of a team, we need to develop a better understanding of how this fundamental process affects the production of the many FOSS technologies we have grown to depend on.

Despite the potential for informing research on distributed work, there is as yet a gap in our understanding of conflict in FOSS teams (Crowston et al., 2012). The limited studies performed to date have shown conflict to be common (Weber, 2005), an important process in project success (Jensen & Scacchi, 2005) and capable of impacting overall team structure (Elliott & Scacchi, 2003).

However, past studies are largely descriptive, focusing on a few case studies of conflict resolution in relatively similar well-established projects with strong Free Software ideals. More work is needed to unpack a more general theory for conflict across different kinds of FOSS projects. At the same time, some important work has been done in similar collaborative settings, such as Wikipedia. However, conflict is often treated unidimensionally, that is, only one aspect is examined at a time and definitions are not always consistent. For instance Collier and Bear (2012) defined conflict as criticism of others when investigating its impact on female contributor retention, while Kittur, Suh, Pendleton, and Chi (2007) examined conflict in the form of reverts across articles, and Matei and Dobrescu (2010) viewed conflict as a more general unproductive form of discussion. Work by Arazy and colleagues (2011; 2013) is a crucial exception, as it draws on theoretical work from conflict in traditional organizational virtual teams to frame its multidimensional understanding of conflict in Wikipedia.

It is important to synthesize these diverse perspectives into a more formal theory of conflict that is relevant to the peer production context more generally, and FOSS specifically. It is also important to use consistent measurement tools and frameworks to allow comparison of group processes across different organizational types, thus developing a more rich understanding of distributed work.

The present work asks the following research question:

How is conflict manifested and interpreted by FOSS project members? What impact does this have on the relationship between conflict, team structures, emergent states and team outcomes?

Highly distributed and ongoing virtual groups are different in several important ways from temporary virtual teams in traditional organizations, and thus the manifestation of processes like conflict can also be expected to vary. For instance, ongoing and distributed virtual groups are more likely to experience membership fluctuations compared to temporary virtual teams (Saunders

& Ahuja, 2006), thus it is important to take into account potentially different conflict experiences of newcomers and more established group members. Furthermore, ongoing virtual groups like FOSS teams are more likely to be involved in recurring task activities(Saunders & Ahuja, 2006), such as incremental and continuous improvement on the same code base, and therefore conflict in past iterations may impact present and future work.

At the same time, though individual member tenures may be finite, as a whole, the group may spend an indefinite time frame working on a piece of software, so long as the software's function continues to satisfy a need. Therefore distributed and ongoing virtual groups like FOSS projects have a stronger sense of context - that is, both a more extensive history together, as well as an expectation of future interaction for a more or less unspecified length of time as compared to short-term virtual teams in traditional organizations. This shift in time frame from a specific end point to an indefinite relationship dramatically changes the incentive structure of a group. An expectation of future interaction shifts some of the group's focus away from just performance, and toward process efficiency, member satisfaction and group identification (Axelrod & Hamilton, 1981; Saunders & Ahuja, 2006). As a result, we may expect conflict in dynamic ongoing distributed work groups such as FOSS projects to be relatively less task-centric, and more focused on improving relationships, procedures, and higher-level group structures and norms.

These are important issues to examine in the FOSS context to further our understanding of how conflict as a process may vary due to different forms of organization in distributed work settings (Mannix & Jehn, 2003). Thus the present work addresses the following research question:

How does conflict as a process differ in dynamic ongoing virtual groups like FOSS projects compared to conflict in short-term virtual teams in traditional organizations?

The importance of studying teams with sufficient shared group history and

expectation of future interaction is particularly relevant in the peer production context, where most interactions are automatically recorded by communication software, archived and inherently more visible (Dabbish, Farzan, Kraut, & Postmes, 2012; Bolici et al., 2009). This serves to broadcast both pro-social and conflictual team member behavior, while adjusting expectations of the whole group about future interactions and allowing other team members to participate in the discussion. Therefore, due to the relatively permeable boundaries of FOSS teams, conflict may attract relatively more attention, particularly among parties not originally involved in the discussion, compared with closed virtual teams in traditional organizations. Emerging issues may intensify due to the formation of coalitions of supporters and become a central feature of the group's discussion (Coser, 1957; Jehn, Rispens, Jonsen, & Greer, 2013).

At the same time, dynamic and ongoing virtual groups with a long term orientation need to adapt to changes in their environment over time. However, they may find their adaptation problematized by their high geographical distribution and reliance on computer-mediated communication (Cramton, 2001). More work is needed to understand how distibuted work groups adapt to changing circumstances, as well as the potentially reciprocal relationships between team processes like conflict and team input structures (Gilson et al., 2014; Martins & Schilpzand, 2011).

While many theories exist on the specific role of conflict in team development (to be reviewed in Chapter 2), they all agree that conflict is a fundamental part of the team lifecycle (Coser, 1957; Gersick, 1988; Tuckman, 1965). Specifically, conflict helps to both revitalize existing norms when they become out of sync with actual team practices, or contribute to the emergence of new norms (Coser, 1957; Packer, 2007). Because group norms set expectations of team members about future interactions, in influencing group norms, conflict becomes a mechanism of adjustment to new conditions (Coser, 1957). Conflict can also be a critical signaling factor for changed conditions that surfaces

issues and allows a team to restructure both socially and technically (Fuchs, 2008; Kelty, 2008). This is particularly true of distributed work that is more dynamic, loosely structured and autonomous (Wageman et al., 2012). Such self-organizing systems are able to reproduce themselves, their own logic and structures (Fuchs, 2008). At the same time, their social dimensions, such as interpersonal processes like conflict, and technical dimensions such as communication technologies, mutually interact, reproduce and shape one another (Jones & Karsten, 2003; Giddens, 1986).

Through an exploration of conflict in the FOSS environment, the present work provides a context for examining the forces involved in the adaptation of distributed work groups to their virtual environment and communication tools (DeSanctis & Poole, 1994). Specifically, the present work will ask the following final research question:

How does conflict impact structural change in dynamic and ongoing virtual groups like FOSS projects?

This dissertation will argue that even though FOSS projects often defer activities that require more interactivity, and rely on implicit rather than explicit coordination practices (Howison, 2009), conflict plays a key role in making important problems more explicit, thus enabling their discussion and resolution, as well as crystallizing and clarifying project norms and influencing the adaptation of group structures over time.

1.1.3 Contribution

Taken together, the present research agenda contributes to the understanding of conflict across three broad areas of inquiry. Firstly, in exploring how conflict as a process is manifested in FOSS teams, the dissertation unifies earlier work on conflict in online collaborative systems and provides a framework for future studies investigating this phenomenon. Using a consistent and relevant framework will help researchers to contrast findings and develop more sophisticated

models to understand group processes. Furthermore, in examining conflict and its relationship with team inputs, emergent states and outcomes, the dissertation provides practical recommendations for identifying and managing unproductive conflict within FOSS teams across different structures and contexts.

Second, in exploring conflict in the context of real-world ongoing virtual groups, the dissertation closes a theoretical gap in existing knowledge on distributed work. Specifically, the present work identifies key paradigms within research on ongoing collocated teams and short-term virtual teams, discusses their relevance to the ongoing virtual group context and draws on them to expand theory on conflict in distributed settings. In doing so, the present dissertation bridges the different areas of inquiry, both enriching work on ongoing virtual groups and delineating their boundaries.

Finally, the work sheds light on the different manifestation of conflict in ongoing virtual team settings as compared to short-term virtual teams. Doing so helps to clarify inconsistencies in findings across earlier studies, in particular with respect to the impact of different types of conflict on emergent states and outcomes in more long-term oriented virtual groups. More importantly, by exploring how conflict as a process differs under conditions of expectations of future interaction and longer time frame, the dissertation contributes insight into the team mechanisms that modify group norms and serve as input for future social and technical structural changes. The present work thereby updates and reformulates the existing conflict framework to take into account ongoing groups' greater focus on group well-being and member support, and conflicts arising from these issues.

1.1.4 Methodology

To address this research agenda, the present dissertation will present two related studies.

The first study (presented in Chapter 3) is an explorative examination of

conflict in FOSS teams through a series of interviews with 22 diverse representatives of the community. A purposeful sample of developers is gathered from different types of projects (from libraries, to end user applications, and complex operating systems) and representing different levels of experience with the team (from leaders and founders, to maintainers, and new contributors). This study uses a grounded approach to understand the manifestation of conflict from the perspective of FOSS participants, and its role in team function. The approach allows a conflict framework to emerge that is meaningful and relevant to FOSS teams specifically, and ongoing virtual teams more broadly.

This work takes a critical step in understanding the types of behavior or issues that are interpreted as conflict by participants, and contrasts this with our existing knowledge of conflict dimensions used in virtual team research thus far. The study further combines this with participant observation and secondary data sources, such as archived communication logs, to trace conflict episodes highlighted by participants over time, and observe their impact on the evolution of group norms, as well as other social or technical structures. In doing so, Study 1 uncovers a new conflict dimension, as well as structures that promote or ameliorate conflict emergence. Specifically, Study 1 uncovers four distinct conflict dimensions (task, affective, process and normative conflict) and four conflict antecedents that impact its emergence in FOSS projects (level of interdepenence in the project, leadership style, extent of geographical distribution and the distribution of decision-making among developers). (Material from Study 1 has appeared in Filippova & Cho, 2015)

In this way the first study sets a framework and foundation for the second study (reported in Chapter 5). Study 2 focuses on generalizing the findings from the first study through a wide survey of randomly sampled FOSS developers. Specifically, through a series of pilots, the study first develops new measurement tools, then reformulates and validates the existing conflict framework to include the new dimension uncovered in Study 1. Following this, using 228

complete responses and a combination of structural equation modeling and hierarchical multiple regressions, the second study quantitatively examines the updated conflict taxonomy. The study measures the differential impact of each conflict type on emergent states such as identification with the team and performance, as well as the intention to remain as a contributor to the project. At the same time, Study 2 traces the impact of structural inputs identified in the first study toward the occurrence of different types of conflict, and investigates possible complex mediating and moderating effects in the relationship between the structural inputs (such as team interdependence, geographical spread, leadership style and empowerment), updated conflict dimensions, emergent states (identification and performance perception), and team vitality. Thus the study contributes a generalized understanding of the interaction of conflict across multiple levels within ongoing virtual teams. (Material from Study 2 is forthcoming in Filippova & Cho, 2016)

1.1.5 Thesis Structure

The remainder of this dissertation will be structured as follows. The rest of this chapter will provide more background on the FOSS phenomenon, its importance and relevance to the study of virtual teams. Literature on the FOSS phenomenon will also be reviewed, in particular lessons learnt from past work on conflict in open collaborative systems, to identify research gaps.

Chapter 2 will follow, tracing the body of thought on conflict in organizations. The chapter will review work on conflict in both traditional (collocated) and virtual teams, while contrasting short-term and long-term team findings. Chapter 2 will further review theories of organizational change, and place conflict as an important process in this space. The chapter will also highlight work on computer-mediated communication, in particular, discussing how the ability of virtual groups to develop relational communication and modify their own structures when given enough time impacts our understanding of conflict in this

setting. In this way the chapter brings together the various threads of research on conflict across different contexts, and identifies a need for greater attention towards the simultaneously positive and negative role of conflict in long-term team development. Chapter 2 will outline broad research questions that underpin two related studies.

Chapter 3 will unpack these broad research questions into concrete research aims. Following this, the chapter will present the methodology of the first explorative study, the study findings as well as a discussion of the findings as they relate to the specific aims of the study.

Chapter 4 will begin with a review of literature based on the four conflict types and input variables identified in the first study. The review will formulate concrete research hypotheses concerning the different relationships between the four conflict variables and emergent states, such as team identification and performance perception, as well as outcomes, namely the intention to remain in the project. Chapter 4 will also draw on a review of prior work to formulate research hypotheses concerning the relationship between structural inputs and the four conflict variables identified in the first study.

Chapter 5 will then present the methodology for the second study, including an overview of the development and reformulation of an updated conflict scale. Chapter 5 will present the study results, and conclude with a discussion of the findings with reference to the concrete research aims of the second study.

Chapter 6 will combine findings from both studies, and discuss them with reference to prior work in the field as well as the broader research questions of this study. In particular, the chapter will discuss in detail how the findings contrast with prior work on conflict in short-term virtual teams, the relevance of findings to broader work on intragroup dynamics, and the implications for our understanding of virtual work.

Chapter 7 will conclude the thesis with an overview of the work, its major findings, and importance to the study of peer production and virtual teams. The

chapter will also outline the specific limitations of both studies, including highlight areas in the design of the studies that are designed to address each other's limitations, and the limitations of the work as whole. The chapter will conclude with recommendations and a path for future research into conflict in ongoing virtual teams.

1.2 Literature Review 1 - Free and Open Source Software

Before progressing with the rest of this work, it is first necessary to briefly introduce the Free and Open Source Software (FOSS) phenomenon, its fundamental organizational principles and distinctions from traditional forms of organization, thereby setting the stage for future discussion. The rest of this chapter will give a brief history of the movement and review literature concerning the phenomenon. Though past research has made significant advances in understanding the successes of FOSS development, including motivations, performance, and intra-group social dynamics, it is also important to recognize that FOSS development does not occur without hindrance. In fact, disagreements and conflict are common. The section will therefore also review literature that has considered conflict in FOSS to-date, and argue there is an unanswered need for a relevant and consistent framework to understand the impact of conflict on FOSS production.

1.2.1 Background and History of Free and Open Source Software

The central premise behind work labeled Free and Open Source Software is that source code should be freely accessible, distributable and modifiable (Weber, 2005). The source code for a program is like a food recipe: it contains human readable instructions on all aspects of what the final product will look and feel like. However, just like a recipe has to be followed to make a final product, source code does not allow a user to run the software without this code first being compiled into binary, or machine readable form. Software is typically distributed as binaries to allow the user to easily consume the product. Binaries are like prepared food, consumable but one cannot (without a great deal of effort) determine how they were put together. Software in which all aspects of

the source code are not visible or accessible to the end user is called proprietary software (Stallman, 2002). The FOSS approach to software development argues that source code should be made available to the end user. However, there are two distinct threads of arguments as to why this should be so.

The Free Software movement believes that software should be free, not necessarily in terms of monetary value, but free the way speech ought to be (Stallman, 2002). The movement believes users should have access to the source code of the software to understand how a program is built. This has implications for personal security as the Free Software movement believes that users have the right to know what information their software consumes (Stallman, 2002). The Free Software movement also believes in the freedom for users to modify the source code as they choose and redistribute modifications, as this allows an individual to make incremental changes for their own use and share them with the world (Stallman, 2002). The Free Software Movement has created a copyleft license that guarantees these freedoms for users. The license, the GNU GPL (Gnu is Not Unix, or GNU, General Public License), grants (rather than restricts) user access to modify and redistribute source code. One of the most famous projects to be released under the GNU GPL License is Linux, created in 1991 by Linus Torvalds (Moon & Sproull, 2002). For this reason Linux is often connected with the Free Software Movement; however Linus is very clear that he does not subscribe to all of the Free Software Foundation ideology (Staff, 2006).

Others shared Linus' reservations, which led to the articulation of the Open Source Software Initiative (OSI) by Eric Raymond, developer of the Sendmail program and well known for his commentary on the FOSS phenomenon (Raymond, 1999). The Open Source Software Initiative is less prescriptive than the Free Software Movement, focusing on the utility of open access to source code, such as the availability of diverse contributions, and the fact that many eyeballs help in finding and solving bugs. Raymond (1999) argues that

this leads to more stable and robust software, as compared to the proprietary model. Though ideologically different, both approaches share the goal of keeping source code open and accessible, and therefore the umbrella term Free and Open Source Software development (FOSS) is used here. This ideology is an important aspect of FOSS projects as its strong normative component creates distinct practices regarding how software should or should not be created, and can be a source of fundamental disagreement within the community.

1.2.2 Past Research on FOSS

Previous work on FOSS can be broadly classified into three categories: inputs, outputs and mediators such as intragroup processes and emergent states. This is a useful classification commonly employed previous work on FOSS (for example Crowston et al. (2012)) and adapted from literature on organizational teamwork (Hackman, 1987; McGrath, 1984). Together, the three categories cover the life cycle of a working group, with processes and outputs sometimes feeding back into inputs in subsequent cycles (Ilgen, Hollenbeck, Johnson, & Jundt, 2005). The following sections will review key work on input, outputs and moderators, in order to highlight a gap in our understanding of conflict as a process.

Inputs

Inputs represent the compositional characteristics of a team, or those factors that are present prior to team membership and may influence group outcomes (Martins, Gilson, Maynard, & Martins, 2004). Though inputs were a key focus of early FOSS research, they continue to be of interest today. Specifically, scholarship aims to uncover structural differences between projects, the type of people who voluntarily participate in open collaborative projects, why they do so and how this may be encouraged.

Early work featured comprehensive surveys on demographics and found a

surprising level of homogeneity among FOSS contributors, with similar age, gender and cultural backgrounds (R. A. Ghosh et al., 2002; Robles, Scheider, Tretkowski, & Weber, 2001). For instance, early studies found FOSS development was highly skewed towards male participation: less than 2% of developers surveyed were female (R. A. Ghosh et al., 2002; Robles et al., 2001). A more recent survey showed incremental improvement a decade later, with approximately 11% of developers identifying as female (Arjona-Reina, Robles, & Dueñas, 2014). Furthermore, most developers surveyed were employed adults in their 20s to 30s for whom FOSS engagement was a part-time activity rather than full time job (Arjona-Reina et al., 2014; R. A. Ghosh et al., 2002; Robles et al., 2001). Studies also showed that a majority of developers across projects surveyed originated from 'Western' countries: over 80% of developers reported being from North America or the European Union, and English was the most commonly spoken language (R. A. Ghosh et al., 2002; Robles et al., 2001). Furthermore, a number of developers reported being involved in several projects simultaneously (R. A. Ghosh et al., 2002), while overall project tenure (involvement in the same project) was relatively low, with 60% involved for under a year (Feller & Fitzgerald, 2002).

Given the high turnover and the fact that most contributors divide their time between full-time jobs and FOSS development, a growing amount of research became concerned with understanding developer motivations for participation. They found that not all participant motivations were purely altruistic: a recent survey shows 38% of developers receive some form of direct financial compensation for their work (Arjona-Reina et al., 2014). Though it is important to dispel the myth of a purely altruistic community, there are also other less tangible motivations for the remaining 60% of contributors. They involve a combination of individual and collective motives. Individual motives include perceived personal benefits derived from participation, such as reputation (Hars & Ou, 2001), visibility to potential employers (Arjona-Reina et al., 2014), in-

trinsic enjoyment (Nov, 2007) and the desire to learn and enhance one's skills (Baytiyeh & Pfaffman, 2010). Collective motives on the other hand involve factors such as altruism as well as the desire to help others (Baytiyeh & Pfaffman, 2010). Wagner and Prasarnphanich (2007) examined individual and collective motives together and found that participants do indeed have mixed motives for participation, but that collective motives outweigh individualistic ones.

Overall, these results show that while FOSS developers derive personal rewards from participation, they are also very much driven by collective goals and values because the success of FOSS projects they are involved in also reflects their personal success.

Outputs

Outputs refer to the consequences of a team's effort, such as the effectiveness of a team, or team performance. Research into FOSS outputs has been largely concerned with how to measure success as well as identifying the various factors that contribute to project performance.

As FOSS development is largely voluntary and exists outside of traditional organizational frameworks, typical performance measures of software production are not appropriate (Wu, Tang, & Pacis, 2007). For example, profit cannot be used as a performance measure, due the nature of licenses used in FOSS creation that explicitly allow free reproduction. Though some projects obtain revenue from supporting services such as enterprise level support, many projects are not financially motivated at all (Weber, 2005). Furthermore, while some projects have fixed release schedules and targets, this is more common in more established projects, and the management of these release schedules varies dramatically based on project aims (Erenkrantz, 2003). In fact, Raymond (1999) argues that rather than sticking to a fixed schedule, it is important to 'release early, release often', as this allows others to benefit from incremental changes more quickly. Such a floating measure cannot be used for the analysis of per-

formance. Finally market share cannot readily be used as a success measure because not all software is built for a market of end-users. Rather it is the result of 'scratching a developer's itch', that is, solving a problem personally relevant for the developer (Raymond, 1999). As such, research into FOSS success needed to devise measures of output appropriate for the context.

Outputs in the context of FOSS development therefore focus on evaluating objective software quality (Chou & He, 2010; Colazo & Fang, 2010; K. J. Stewart & Gosain, 2006), developer output (Colazo & Fang, 2010; Xu, Jones, & Shao, 2009) as well as community/external impact in the project and popularity with end users (Crowston, Annabi, & Howison, 2003; Midha & Palvia, 2012; Sen, Singh, & Borle, 2012). Studies also evaluate subjective measures like developers' perceptions of quality and performance (Ke & Zhang, 2011; S. T. Lee, Kim, & Gupta, 2009) and their satisfaction with contributing to the project (Casaló, Cisneros, Flavián, & Guinalíu, 2009). Though the material means of FOSS production are significantly different from closed source software development and other organizational teams, the above output measures suggest it is still possible and relevant to understand FOSS outputs through general measures adapted from traditional organizational contexts.

In exploring FOSS outputs, research has been concerned with understanding the kinds of structural inputs that result in successful FOSS projects. For example, Sen et al. (2012) used project characteristic measures to uncover what attracts developers to projects. They found that participation is positively connected with factors such as the use of the C programming language for development and less restrictive licenses that allow commercial use. This is logical as C is a commonly used language often studied in educational institutions, and software licenses that do not enforce releasing derivatives as fully open sourced (less restrictive) cater to many more applications (Subramaniam, Sen, & Nelson, 2009).

However, structural input factors, such as the choice of license and program-

ming language, are often hard to modify after the initial stages of a project. For research aiming to understand how to learn from and improve FOSS team performance, it is particularly instructive to examine more dynamic moderating factors to understand how they may be employed to encourage participation and positive outputs, and when they detract from project success.

Processes and Emergent States

While research has examined both technical and social mediators between FOSS structures and their success, the social mediators are of primary interest to the present work. Social mediators, in turn, can be separated into processes and emergent states (Marks, Mathieu, & Zaccaro, 2001). While processes involve the interdependent activities of team members that convert inputs into outputs, emergent states are properties of the entire team that are dynamic and vary as a function of context, input, processes and outcomes (Marks et al., 2001).

The process of FOSS coordination, and its impact on outcomes, is of particular interest to research on the phenomenon. Research on both FOSS development and other forms of open collaboration consistently finds a power-law participation pattern, in other words, that a majority of contributions are performed by a small percentage of team members. For example, Moon and Sproull (2002) find that in the Linux Kernel mailing list, 2% of participants contribute more than 50% of messages. Of these, almost half are recognized as formal members of the project. Wasko and Faraj (2005) similarly find that a core critical mass of active and closely-knit participants is necessary to sustain such collaborative groups. Crowston and Howison (2006) elaborate that this developer core exists at the heart of an onion model of role differentiation, with each successive outward layer lower in responsibility, but more populous than the inner layers.

Given that a small number of developers are responsible for a big portion of the outcomes, studies are also interested in how this critical mass forms and how it can be sustained. In a longitudinal analysis of the evolution of 147 FOSS

projects, Long and Siau (2007) discovered the tendency of projects to start out as a single hub and expand into a core/periphery model over time. Amrit and Van Hillegersberg (2010) found that a steady flow of members from the periphery to the core was indicative of a healthy project, whereas a steady outflow from the core indicated a project that was having coordination problems. It is important to understand therefore what encourages individuals to work towards joining the core and prevents them from leaving.

Research has tried to understand how members progress towards more formal roles by examining how members become socialized. Duchenaut (2005) finds that successful integration involves rites of passage, the construction of an identity consistent with group norms, and the ability to navigate internal politics by building a network of relationships. Tullio and Devan (2008) similarly show that socialization occurs as individual sense making comes into contact with and aligns toward shared community meanings. Furthermore, Qureshi and Fang (2011) studied the successful movement of peripheral members into core teams and found this process to vary with levels of initial and sustained socialization. Peripheral members who socialized highly and continually with other project members were integrated into the core significantly faster (7.5 weeks) than members who socialized less (up to 83.2 weeks). Taken together, all the above studies suggest that getting more familiar with and aligning toward shared team social norms is a critical component of new member socialization.

The relative level of alignment toward group norms and community values, or team identification, is a critical emergent state in FOSS teams that is consistently connected with positive team outcomes such as performance, satisfaction with the team and participation intention (Bagozzi & Dholakia, 2005; K. J. Stewart & Gosain, 2006; Xu et al., 2009; Zhu, Kraut, & Kittur, 2012). In fact, Fang and Neufeld (2009) have found that situated learning and identification with the group were stronger predictors of intention to remain in the project than initial motivations to join.

Given that projects adopt more complex social structures over time, and there appears to be a key core of individuals that new members strive to become a part of, research is also interested to understand governance in this context. Specifically, studies find that projects vary widely in their degree of centralization and authority structures, and that this is an emergent quality of teams over time. On one hand, Mockus, Fielding, and Herbsleb (2002) find that in the Apache project, a decentralized decision making structure is in place that involves a simple voting process. On the other, Jensen and Scacchi (2005) document a much more formalized structure in the Netbeans community. Most projects exist on a continuum between these extremes (Crowston, Kangning Wei, Qing Li, & Howison, 2006), and O'Mahony and Ferraro (2007) suggest this is because leadership structures emerge over time in FOSS projects depending on various contextual factors and team aims. Giuri, Rullani, and Torrisi (2008) further find that more modular projects tend toward more pronounced leadership structures, while leaders emerge from participants with more diverse skill because this helps to maintain better overview of different aspects of development. Thus different governance models appear to be effective for projects with different contexts.

Knowledge sharing is another critical emergent state for FOSS teams (Hemetsberger & Reinhardt, 2004). The use of virtual communication tools to archive and record interactions supports FOSS team longevity by enabling transactive group memory and the re-experience of events and decisions by new group members, affording persistent guidance in the face of high turnover.

Overall, studies of the social processes behind the different levels of membership in FOSS development highlight the importance of cohesion, good standing within the group and strong interpersonal relationships with other members of the group, to the effectiveness of FOSS teams. However, past research on FOSS processes was largely interested in understanding what makes projects successful, and situations in which developers work together well. Relatively

fewer research to-date has considered instances of disagreement between developers, and their implications for FOSS projects.

Past work on Conflict in FOSS

Research on FOSS development suggests that conflict is very much prevalent in community interactions (Bezroukov, 1999), and integral to development functions (Weber, 2005). However, as the next section will show, significant gaps remain in our understanding of this process. Because conflict can be both detrimental to a group's function, as well as capable of encouraging novel thinking and innovation (Coser, 1957), studying conflict is critical to the understanding of both the successes and failures of open collaborative systems (Crowston et al., 2012).

A number of early studies that dealt directly with conflict were based on case studies of individual projects, and thus have taken a more descriptive approach. For instance, Elliott and Scacchi (2003) described an example of conflict that emerged in a Free Software project developing a business office system (BOS). The study highlighted disagreements arising from a perceived inconsistency between the project's stated goals of creating Free Software, and the implementation of these goals by the use of non-free (proprietary, non-open source) tools. The conflict between freedom ideals and non-Free software use resulted in a clarification of community norms that allowed developers the freedom to use any (including non-free) tools to create Free Software. An interesting insight emerging from this work suggests that conflict episodes may influence group structure and principles when group norms are called into question. Recent work by Wang, Shih, and Carroll (2015) supports these findings by studying value diversity more generally in a case study of the Mozilla project. They find that when opinions differ concerning the goal of the project/community, this may outwardly manifest in conflict.

The above studies show an interesting pattern of conflict emerging from clar-

ifications of broad community goals and norms. However, the above studies are largely descriptive, do not explicitly provide a consistent definition of conflict, or contrast their findings with extant literature. Thus it is difficult to draw comparisons and understand if there are significant differences in experiences of FOSS projects compared with other forms of organization.

Van Wendel de Joode's (2004) work is one exception that engages with a conflict framework that is derived from work on traditional organizations and commonly used in virtual team research. The study distinguishes between conflicts that arise from tasks disagreements concerning software development and affective disagreements that arise from interpersonal incompatibilities. However, the norms-related disagreements examined above do not fit into the distinction made by van Wendel de Joode, suggesting the need to expand this task/affective conflict binary.

Other work on FOSS conflict deals directly with conflict management. For example, Jensen and Scacchi (2005) look at conflict resolution in the Netbeans community and find that conflicts are resolved primarily via discussion mailing lists, thus allowing the entire community the opportunity to weigh in on the issue. van Wendel de Joode (2004) also finds that modularity and ability to develop branches of code in parallel, or in extreme cases, fork a project, are a potential means of managing conflict in FOSS teams. While conflict management is an important area of study, not all types of conflicts are necessarily detrimental for team development (Amason, 1996; Coser, 1957). Thus it is also pertinent to distinguish types of conflicts and conditions under which they may or may not need resolution in FOSS settings. This requires a consistent theoretical formulation of conflict occurrence in FOSS.

Though FOSS studies on conflict are limited, work on conflict in other open collaborative systems, like Wikipedia, is somewhat more prevalent and may be instructive. Thus key research in this area is also reviewed below.

Past work on Conflict in Wikipedia

Kittur and colleagues (2010; 2009; 2007) were some of the first to consider what conflict means in a Wikipedia setting. In their work, conflict is conceptualized as the level of reverts in an article, or the amount of times changes of some participants have been undone by other contributors. In an earlier study, they develop a method for predicting reverts from article content (Kittur et al., 2007). Following this, they find an increase in the number of contributors increases conflict levels due to the presence of more varied perspectives (Kittur et al., 2009). They further find that modifications to policy and greater procedural work are the primary means for resolving these disagreements.

Arazy et al. (2011) build on these findings and connect them with conflict theory on traditional organizational teamwork. Specifically, they examine the emergence of conflict due to differing opinions on the task at hand, as an example of task conflict. Alongside this, they examine the level of cognitive diversity among authors of the same Wikipedia article (representing a team), and their connection with the quality of articles produced. They find that on its own, task conflict has a negative relationship with article quality. However, when it is present in groups with greater cognitive diversity, task conflict enables these varied opinions to surface and contributes to greater article quality. In a subsequent study, Arazy et al. (2013) also find that task conflict among Wikipedia editors can evolve into procedural or interpersonal issues if left unresolved, and that this transformation has a negative impact on the quality of articles written.

Similar to van Wendel de Joode's work reviewed earlier, the above studies on Wikipedia suggests a trend toward drawing on conflict theory from traditional organizational settings to improve our understanding of open collaborative system processes. However, also similar to work on FOSS, other studies on Wikipedia conflict suggest there may be more to conflict in open collaborative settings than the traditional conflict framework covers. For instance, Matei and Dobrescu (2010) find that conflict emerges from disagreements concern-

ing group norms, and overall group values, such as Wikipedia's neutral point of view (NPOV) policy. This type of conflict is fundamental in enabling the surfacing of different interpretations of intentionally vague group norms, and results in the development of a common understanding between contributors. In this way, similar to work on FOSS conflict reviewed above, Matei and Dobrescu show that conflict can arise from unclear group norms and in its resolution be instrumental in the development of the community as a whole.

It is important to recognize that despite the many similarities, work to edit Wikipedia and create FOSS projects has some important differences that may impact theory building on conflict. Aside from differences in the subject matter of the work (namely software and encyclopaedic knowledge), editing work in Wiki settings affords less task interdependence than FOSS development because only one editor can edit an article at a time. This results in the need for more discussion about the right way to go about editing an article on the article talk page, and may lead to more explicit manifested disagreements. Most crucially, Wikipedia affords less opportunity for attaining visible reputation in the community compared to FOSS teams who reserve special titles for core contributors and highlight the importance of many central project members on project pages.

Thus motivations of Wikipedia editors and FOSS developers differ: FOSS developers are motivated both by individual (more selfish) reasons such as reputation in the community as this aids in career advancement offline, as well as by collective goals such as belonging to a project and a belief in the ideals of Free and Open Source Software (Oreg & Nov, 2008). By contrast, Wikipedia editors are more likely to contribute for altruistic reasons alone, and their motivations mainly depend on their belief in the need for continuing to develop an open and transparent source of knowledge (Nov, 2007). The implication of these differences is that conflicts may have an event stronger effect on Wiki-like communities than might be observed in the present study on FOSS teams. This is because in the absence of personal motivations to contribute to the project,

Wiki editors may find less value in continuing to contribute in the presence of conflict and therefore more likely to depart the community.

Keeping in mind the potential for conflict to be even more destructive in Wikipedia settings, we can derive some common patterns and insights from studies on conflict on FOSS and Wikipedia. For instance, in attempting to build more formal theory on conflict, work in both spheres draws from literature on traditional organizational teamwork. However, studies also suggest there may be more to conflict than disagreements about task, procedure, or interpersonal incompatibilities. Specifically, work suggests conflicts about group norms play an important role in team development and shape, crystallize and modify subsequent community values and beliefs, suggesting the need to build theory to include this dimension in our understanding of conflict. Thus the present work takes first steps toward building a consistent and relevant theory of conflict for FOSS teams with the following research question:

How is conflict manifested and interpreted by FOSS project members? What impact does this have on the relationship between conflict, team structures, emergent states and team outcomes?

In addressing this question, following prior work on open collaborative systems, the present research will draw on literature from organizational teams to help understand how this evidence of conflict about norms fits into our understanding of intragroup conflict more broadly. Before moving on to a review of conflict conceptualization in traditional organizational settings, the next section will consider to what extent it is appropriate to draw from literature on traditional organizations to build theory on conflict in an emerging online collaboration phenomenon.

1.2.3 Relevance of organizational theory to FOSS research

As the above section has shown, work on open collaborative systems often borrows from literature on virtual teams to frame their enquiry (e.g. Arazy et al.,

2011; Crowston et al., 2007; Moon & Sproull, 2002). This is because FOSS teams have several properties in common with virtual teams. These are described below, and an argument is made for drawing parallels between virtual team research and FOSS when building an understanding of conflict.

Virtual teams are "groups of people with a common purpose who carry out interdependent tasks across locations and time, using technology much more than they use face-to-face interactions" (Cramton, 2001, p. 346). The definition of virtual teams, therefore, contains within it three distinct variables: 1) common purpose, 2) interdependence of task and 3) technology as primary means of communication. FOSS developers do indeed work together on a common purpose, and as face-to-face contact between developers is rare and often unplanned, they use technology as a primary means of communication (Crowston et al., 2007).

FOSS teams are also interdependent, though in somewhat different ways from virtual teams in traditional organizations. As Howison (2009) notes, tasks in FOSS development tend to be modularized, which means that development of any module can occur relatively independently from the rest. Howison (2009) points out that in FOSS development, tasks that require substantial effort to perform, such as those needing to involve more than one individual, are frequently deferred until such a time when the code base changes sufficiently to allow the task to be performed individually. Furthermore, coordination happens through implicit mechanisms such as extensive logging, the use of common tools and a preference for small, incremental changes (Bolici et al., 2009). These features, together with open contribution policies and licenses allow for technical independence during development.

However, there are other levels of social interdependence experienced by collaborators in FOSS teams. For instance, many projects employ a system of access management that limits who can make changes to the main branch of source code (Weber, 2005). This is necessary to avoid a situation in which a

commit breaks functionality of the system. Getting commit access that allows developers to bypass this and make changes directly to the code base often requires some efforts in socializing into the community (Duchenaut, 2005), and even after attaining this status, committers remain bound by social norms like code review and subject to comments from other community members. Furthermore, code contributors without commit privileges need to coordinate their modifications with the maintainers (von Krogh, Spaeth, & Lakhani, 2003). Thus while development of individual pieces of code can occur independently, in larger projects this code is still subject to approval by other community members, maintainers or the leader, before it can be merged. Therefore most FOSS team members depend on other contributors to achieve their goals, just like virtual teams, but the manifestations of interdependence vary in practice.

Finally, FOSS teams address the problem of division of labor by affording voluntary participants the chance to select tasks they work on. In other words, most contributions are the result of "scratching a developer's itch" rather than a planned set of tasks (Raymond, 1999). However, though various factors unique to FOSS enable this mechanism, on the whole it is not an entirely new idea. Work on traditional organizations prior to the emergence of virtual teamwork has already proposed the idea of team empowerment, that is, the sharing of power between the leader and their subordinates such that team members enjoy greater ability to set their own goals (Conger & Kanungo, 1988). This idea has also been extended to the virtual team research space in the form of shared leadership, or collaborative decision-making (Bell & Kozlowski, 2002; Hoch & Kozlowski, 2014). Supporting the parallel between shared leadership and FOSS, Fielding (1999) examined the distributed and collaborative decision-making style of the Apache project as an example of shared leadership.

Taken together, it appears that while FOSS teams and virtual teams may employ different solutions, these solutions are directed at addressing the same problems. In fact, Puranam, Alexy, and Reitzig (2014) argue that all organiza-

tions address the same key organizing principles concerning division of labor and integration of effort. New forms of organization simply offer novel solutions to one or more of these principles. Crucially, while they may appear to be new for some forms of organizations, they may in fact be novel bundles of older properties or solutions. In this case, rather than developing entirely new theories, it is fruitful to draw on areas of insight where these features have been well studied, to afford greater consistency as well as comparisons across different organizational types.

Thus the present work draw parallels between FOSS projects and virtual teams in traditional organizational settings by examining the extent to which theories of conflict in traditional organizational settings are applicable or may differ in the FOSS context.

Terminology: Groups versus Teams

It is important to note at this juncture the terminology used throughout the rest of the dissertation when talking about FOSS projects. Specifically, throughout this work, the terms "group", "team" and "FOSS project" are often used interchangeably. However, both the terms "group" and "team" are loaded constructs, and it is important to recognize the different perspectives surrounding these terms.

Some prior work emphasizes the distinction between groups and teams. For example, Katzenbach and Smith (1993) suggest that "A team is a small number of people with complementary skills who are committed to a common purpose, set of performance goals, and approach for which they hold themselves mutually accountable." (Katzenbach & Smith, 1993, p. 112) By contrast, groups are more a collection of individual actions, and are not responsible for results other than their own. This is a fruitful distinction in an organizational context, however it does not translate easily to the peer production context, and to FOSS projects more specifically. While FOSS projects are often made up of a collection of individual contributions (Howison, 2009), they also feel connected by a com-

mon purpose, that is, the creation of Free and Open Source Software, and are accountable to one another for the approach they take to solving the problem. In other words, despite often involving very modularized work, FOSS teams share a sense of group identity and have common processes to which others are accountable.

Other research does not distinguish between work teams and work groups explicitly. For instance, Kozlowski and Bell (2001) performed an extensive review of work on the subject and proposed that both work teams and groups are:

- entities composed of two or more individuals;
- who exist to perform organizationally relevant tasks;
- share one or more common goals;
- interact socially;
- exhibit some form of task interdependence, be it in goals, outcomes or workflow; and
- maintain and manage boundaries of membership and are embedded in a broader organizational context that influences the group.

In their definition, FOSS projects represent work groups/teams as well, because they can be comprised of anywhere from two to hundreds of individuals that perform tasks relevant to the project development as a whole, and despite individual motivations share a common goal of contributing to the development of Free and Open Source Software (Kozlowski & Bell, 2001). Furthermore, FOSS project members often interact socially on IRC, mailing lists and during community events, and they have most frequently sequential interdependence in their tasks as described above. In addition, FOSS projects do maintain boundaries and access to project membership because not everyone can become a contributor - one must have sufficient knowledge in the technologies used by a particular project, demonstrate an understanding of the project code base by proposing a valuable improvement to the project and be recognized as such by

other members. Finally, different groups of developers working on smaller sections of the project code base are embedded in the broader project context, as well as the more general FOSS community. Thus a group of developers working together on a Ruby library are also embedded in the broader structure of the Ruby community, any other projects they may depend on, and the FOSS ecosystem as a whole.

To facilitate comparison between different types of work organizations, Kozlowski and Bell (2001) propose 6 dimensions that distinguish different forms of work:

> 1. *"the external environment or organizational context in terms of its (a) dynamics and (b) degree of required coupling;*
> 2. *team boundary permeability and spanning,*
> 3. *member (a) diversity and (b) collocation/spatial distribution;*
> 4. *internal coupling requirements;*
> 5. *workflow interdependence with its implications for (a) goal, (b) role, (c) process, and (d) performance demands; and*
> 6. *temporal characteristics that determine the nature of (a) performance episodes and cycles and (b) the team life cycle"*
>
> (Kozlowski & Bell, 2001, p. 10).

According to this framework, short-term virtual teams in traditional organizations are embedded in a relatively closed organizational context and may be mostly uncoupled from other teams in the organization while having lower team boundary permeability. They do however have relatively high geographical distribution, diversity, internal coupling, and interdependence in work, goals, and performance but more short-term performance and team life cycles. By contrast, FOSS teams have an open and transparent organizational environment that communicates a lot of its own inner-workings to the public and a higher degree of coupling between different groups of developers in the same project

because they work on the same code base and may therefore break each other's code. FOSS projects also have high boundary permeability, diversity and geographical distribution, relatively lower internal coupling requirements within teams and less workflow interdependence, with long-term team life cycles interspersed with repeated performance milestones in the form of software releases. Specifically, FOSS projects can span many decades: Linux, for instance, was first released in 1991 (Moon & Sproull, 2002), the GNU project was announced in 1983 (*Overview of the GNU System*, n.d.) and Debian began in 1993 (*About Debian*, n.d.).

The above framework helps in translating theories from traditional organizational settings into peer production contexts by clearly delineating areas of similarity and stark contrast. Thus when the present work describes FOSS projects as groups or teams, it is with the understanding that they have a unique pattern of work organization that is in some ways similar and in other ways different to that of work groups in traditional organizations. The findings of the studies reported in this work may therefore be generalized more along some of these similar dimensions, but vary across dimensions that are in stark contrast to organizational teams. This is one of the key aims of this work, because in drawing these clear boundaries it is then possible to expand both theory on work in FOSS settings as well as conflict in distributed work by exploring and contrasting new and different organizational contexts. The following chapter will outline this research agenda in more detail, elaborating how conflict manifestations in FOSS projects may differ from those of short-term virtual teams in traditional organizations and the implications of this difference for conflict theory as a whole.

Chapter 2

Literature Review 2 - Conflict in Organizational Teams

This chapter expands theory on conflict in both traditional (collocated) and virtual work settings, and illustrates why it is necessary to examine conflict in ongoing virtual team settings in greater detail. Specifically, this second chapter synthesizes work across various fields to build an argument for expanding the existing taxonomy of conflict in virtual work. First, an overview of conflict conceptualizations within different domains is presented, highlighting the role of conflict in stimulating change within social organizations. Then, by exploring relevant CMC theories, the chapter suggests virtual groups have a similar capacity for internal transformation and growth. A review of existing literature on conflict in virtual work is presented, highlighting recent advancements as well as a gap in understanding of ongoing virtual teams and the processes through which they evolve. Finally, an argument is made for the need to examine conflict in ongoing virtual team settings, and in particular, its relationship with group development and future team structures.

2.1 Classic theory on conflict in social groups

Historically, scholarly work viewed conflict in opposition to order. Early thinkers emphasized the need for maintaining order in social organizations by

controlling dissent and maintaining a strong central authority (e.g. Hobbes, 1651). Conflict was therefore viewed as a destabilization of an inherently stable and functional society (Parsons, 1951), and early organizational literature inherited this dialectic assumption. Conflict was seen as detrimental to cooperation and organizational effectiveness (Mayo, 1933), and research emphasized the need to control conflict as far as possible, by applying the right management principles (Taylor, 1911) and structures (Fayol, 1949).

Taken together, this foundational work presupposed that a healthy social organization is one that is free of conflict. However, other work rooted in the Hegelian tradition argued that conflict is a primary driver for change in a group. Specifically, social change emerges from two opposing forces: the thesis (or the status quo) and the anti-thesis (or the resistance) derived inevitably from and in opposition to the status quo (Beiser, 2005). Growth, or synthesis, only occurs after these two opposing forces come to a head and are resolved in favor of a new status quo. In this recursive process, the status quo is eventually challenged with a new resistance in future iterations.

A certain amount of conflict is essential for holding any social group together – from marriage, to organizations, communities and nations (Simmel, 1955). This is because conflict is a primary driver of growth by highlighting various options for action, while stirring us to reflect and act upon those options (Dewey, 1922). Thus while conflict as a social process does have potential to distract a group from its immediate goal, it also affords an opportunity for new opinions and directions, reflection about existing structures and action towards change.

The next sections will examine how literature on traditional organizational teams has treated conflict thus far with respect to its nature, dimensions and impact on the long-term development of collocated teams, highlighting areas where this dissertation can make a contribution.

2.2 Conflict in organizations

The following section will first outline a definition of conflict as a process in the context of organizational literature. A historical overview of key research trends in this area will be presented, followed by a review of more recent work. Given the large amount of prior work on the subject of conflict, a full review would be impractical. However, the following sections aim to highlight overall tends in the development of knowledge about conflict in organizational teams over time.

2.2.1 Definition

Although early definitions of conflict varied widely and no one definition was predominant, over time research coalesced toward characterizations that included a collection of similar factors: a *dynamic* group process arising from a level of *interdependence* within the team due to the *perception* that one's concerns have been interfered with (whether or not this is actually the case) and *manifesting* in dissonance, disagreement or incompatibility (Fink, 1968).

Pondy (1967) suggested that conflict is a dynamic process between two or more individuals that emerges as a sequence of episodes and gradually escalates into disorder. Conflict begins with certain potentials that may or may not be recognized by individuals. If these potentials are perceived as incompatible, they are likely to manifest in a variety of conflictual behavior, with each conflict episode leaving an aftermath that affects future interactions between team members. Similarly, Putnam and Poole (1987) argued that conflict emerges from the interaction of interdependent people who perceive opposition to goals/aims/values and who see the other party as potentially interfering with those aims.

Others prefer a more simple definition. Thomas (1992) asserts that conflict is "the process which begins when one party perceives that another has frustrated, or is about to frustrate, some concern of his" (p. 891). Rahim (2002) suggests that conflict is an interactive process manifested in incompatibility,

disagreement, or dissonance within or between social entities. Rahim's definition includes conflict across different levels: intrapersonal (internal conflict), interpersonal, intragroup, intergroup and interorganizational.

One aspect the various definitions do not exactly agree on concerns the nature of the opposition perceived in the conflict episode: Putnam and Poole (1987) discuss goals, aims and values; Thomas (1992) talks about concerns; Rahim specifies only a general incompatibility, while J. A. Wall and Callister (1995) also describe interests. These diverse perspectives on the nature of opposition have become reflected in a section of conflict literature that distinguishes between different types of conflict and their varying effects on the team. Furthermore, some approaches to conflict consider the way an episode unfolds, while others instead focus on events and prior causes. The next two sections will introduce these differing perspectives on conflict, beginning with process versus structural approaches.

2.2.2 Process versus structural approaches

Early conflict literature was focused on either exploring the process through which a conflict episode unfolded, or in identifying the structural conditions that led to conflict manifestation. In doing so, the two approaches made different assumptions about the emergence of conflict.

Process approaches

In a prominent example of the process approach, Pondy (1967) treats conflict as an episode that unfolds in a series of stages: latent conflict, perceived conflict, felt conflict, manifest conflict, and conflict aftermath. Latent conflict emerges from a combination of structural conflict potentials, or, in other words, latent conflict is the potential for conflict based on certain structural team characteristics. In the perceived conflict stage, one or more parties have an awareness of conflicting interests. In the felt conflict stage the parties experience tension as a

result of this awareness, and/or personal attribution. Manifest conflict for Pondy involves open aggression, while the conflict aftermath is a legacy of conflict that is not fully resolved and may reappear or be aggravated in future interactions. These stages do not need to occur strictly in sequence. Latent conflict may not always result in perceived conflict or aggression; similarly, parties may perceive conflict even if no latent conditions exist for its basis.

Walton (1969) proposed a similar approach involving conflict cycles that begin with substantive emotional issues that have the potential to trigger conflict events and lead to manifest conflict. Individuals may exhibit different behaviors during the conflict episode, with varying consequences. Thus in Walton's model, the extent to which conflict is a negative force in the team largely depends on team member behavior. The conflict episode also eventually feeds back into new or redefined issues for the team.

Similarly, Thomas (1992) argued that conflict begins with an initial frustration, which leads to differing behaviors depending on the perception of parties concerning the conflict episode. Furthermore, individual behaviors are subject to interpretation by other team members, resulting in a feedback loop that may escalate and become a self-fulfilling prophecy. Thus if one individual perceives a frustration of their goals from another team member (even if this was not deliberate), and reacts with hostility, the second team member may interpret this hostility and react negatively in turn, thereby escalating the event. Finally, similar to the models outlined above, Thomas' conceptualization also suggests that outcomes of the conflict, especially when unresolved, have the capacity to impact and stimulate future frustrations.

Taken together, the above process models share many similarities. All the models assume conflict follows a predictable and recursive course. They all feature an interpretive dimension, in which conflict must be perceived as such before it can manifest in the team, and an emphasis that these perceptions may be different and not necessarily in line with objective reality. However, this

emphasis on interpretation means that conflict need not always have a negative effect on the team – it is only when behavior is recognized as conflictual and elicits negative responses that it begins to negatively impact the team. Furthermore, all the above models agree that conflict has an effect on future interactions by setting up context between parties and social conditions that can exacerbate future issues, especially when conflict is not resolved early.

Structural approaches

Structural approaches to conflict, by contrast, often assume a rational response of team members to certain preconditions. For example, Deutsch (1949), in one of the earliest works on the subject, highlights the significance of goal interdependence for the emergence of conflict. Specifically, when individuals have positively interdependent goals, this results in cooperation, while negative interdependence results in competition and conflict. This perspective draws a clear and direct line between structural inputs and conflict emergence. Research focused on identifying structures that lead to conflict has examined intrapersonal, intragroup as well as contextual and external factors.

Research examining intrapersonal factors in the rational response model highlighted internal attributes that when present, are likely to be connected with greater conflict. These are also sometimes referred to as internal actor attributes, and reflect issues such as individual bias when forming perceptions of the communication intent (Thomas & Pondy, 1977), incomplete information (Hackathorn & Keen, 1981), predispositions towards bargaining styles and varying personal motives (Brett, 1984), as well as position in the organization (Pondy, 1967).

Interpersonal and intragroup factors are some of the most commonly studied, and thus exhibit wide variety across past work. Interpersonal, or dyadic, factors focus on relationship nuances between two opposing parties, such as communication styles and hostile behavior (Pondy, 1967). A number of stud-

ies distinguish between different types of conflict sources such conflicts about task related issues, interpersonal disagreements and procedural issues (Amason, 1996; Behfar, Peterson, Mannix, & Trochim, 2008; Jehn, 1995, 1997). These will be examined in greater detail in the next section. Research also considers the norms of reciprocity between parties in influencing a conflict episode (Gouldner, 1960; Park & Antonioni, 2007). For instance, studies show that while in some situations the most rational response would be for individuals to compete, when they are presented with information about others cooperative intent, they are more likely to reciprocate (Falk & Fischbacher, 2006). Thus when team members expect future interactions with one another, reciprocity reduces competitive drives that lead to conflict. A more generalized group norm of reciprocity has also been shown to encourage knowledge sharing and cooperation in virtual communities at an intragroup level (Wasko & Faraj, 2005).

Power is another important interpersonal factor examined in prior work. Emerson (1962) outlines the power-dependence relationship in which power is a function of the control, or influence, an individual is able to exert and dependence involves the value of the outcome and availability of alternatives. Together these factors determine an individual's rational choice toward cooperation or competition in the face of conflict. For example, if an individual perceives they have relatively low power over their conflict opponent, and a high dependence on them for a personal outcome, they may be more likely to compromise or accommodate the other party. In virtual groups like FOSS teams, the voluntary nature of developer participation may mean that other developers or project leaders have relatively less influence they can exert on their team members in times of disagreement.

Some of the most notable and commonly studied intergroup factors include the level of goal/task compatibility, task clarity and complexity, organization or group norms, as well as team interdependence (Blake & Mouton, 1984; Brett, 1984; Deutsch, 1949; Kilmann & Thomas, 1977). Greater goal or task compat-

ibility among group members and prominent group norms are generally seen to be favorable for the team and reduce unproductive conflict levels. Greater task complexity stimulates more conflict about task issues, but leads to better quality outcomes if task goals are clear, while interdependence often leads to greater levels of both productive and unproductive conflicts (Yetton & Sharma, 2007).

Finally, intergroup and contextual factors deal with issues outside of the group that may stimulate or ameliorate conflict occurrence. For instance, Sherif (1936) has found that intergroup competition and conflict may occur from the simple act of dividing individuals into opposing groups, and setting up competitive goals. However, superordinate goals are helpful in bridging otherwise competitive behavior between two groups, because this increases strength of ties and identification with the team (Nelson, 1989). Context can mean many things in the context of organizational conflict, such as the broader organizational setting, norms, and market position (Pondy, 1967). Context can also include the history of previous interactions group members may have had with each other as members of this or other past teams (J. A. Wall & Callister, 1995).

Pondy (1967) found that among the various structural factors examined by literature, relational variables had the most overall predictive power (such as interdependence, organization position, and norms of reciprocity), followed by conflict issues or sources, with actor attributes and organizational context variables accounting for relatively less variance in predictive models of conflict emergence.

Taken together, though the process and structural approaches to the early study of conflict have somewhat different assumptions, there are a number of similarities in the approaches when viewed as a whole. Both approaches recognize the role of individual perceptions on conflict manifestation, as well as the relevance of past interactions on subsequent conflict emergence. Thus both approaches suggest a need to consider subjective conflict interpretations, and the evolution of conflict over time. Furthermore, both views identify that conflict

may be concerned with different issues, and have varying effects on the team, both positive and negative. In this way, early research has shaped more recent work on conflict in the 1990s and 2000s that focused on distinguishing between different types of conflict, and their differential effect on team outcomes.

2.2.3 Towards an intragroup conflict taxonomy

As the previous section has shown, the recognition that conflict can have both good and bad implications for the team has sparked more attention, particularly in the past two decades, towards identifying the specific types of conflicts and conditions that may lead to these effects. The following section reviews this avenue of research in greater detail, and introduces the intragroup conflict taxonomy used predominantly in conflict research today.

Work as early as the 1960s had already distinguished between destructive conflict and instrumental, goal oriented and rational conflict models (Rapoport, 1960). Similarly, Pondy (1967) argued that conflict could be both functional and dysfunctional. Furthermore, as previous sections have shown, conflict does not necessarily have to be costly, and may be a stable feature of the group. Thus conflict may not always need resolution – this largely depends on the stability and severity of relationships involved (Pondy, 1967).

Early researchers also distinguished between different types of conflicts. Guetzkow and Gyr (1954), for instance, distinguished between conflict grounded in interpersonal incompatibilities and issues arising from the task at hand. Similarly, Rapoport (1960) suggested a difference between fight conflict, an aggressive and destabilizing force, and debate conflict that involved discussions of what is and what ought to be, such as different ideas, values, ideologies or policies. V. D. Wall and Nolan (1986) identified person conflict that focused on relationship differences, and substantive conflict about the task at hand. Pondy (1967), on the other hand, suggested that conflict might arise out of scarce resources, thus problematizing the achievement of the task at hand, or due to dis-

agreements regarding aspects of control such as rules, procedure or leadership, and coordination.

In 1990, Pinkley set out to formalize some of these disputant interpretations of conflict, or the various ways conflict is framed, and isolated three dimensions: task versus relationship, intellectual versus emotional, and win versus compromise (Pinkley, 1990). The first dimension involved the extent to which a conflict episode was attributed to problems in the relationship between parties. The second was concerned with the extent to which individuals paid attention to emotional aspects of the conflict episode, such as feelings of anger and frustration. The third involved attribution of blame, that is the extent to which parties felt that both individuals were responsible for the conflict. Individuals who felt the conflict was a result of the actions of both parties were more likely to seek cooperative outcomes, while those who felt only the other party was to blame were more likely to seek compensation.

Over time, the relationship and affective dimensions became conflated and used interchangeably, as did the task and cognitive dimensions. Task conflict became associated with improving the quality of team outputs by introducing more opinions and preventing groupthink, while relationship conflict, due to its often-accompanying emotional dimension, became associated with a reduction of team satisfaction and performance (Amason, 1996; De Dreu & West, 2001; Jehn, 1995). Amason (1996) and Jehn (1995) both used the relationship/task dichotomy to test these propositions. Amason found that task conflict had an overall positive relationship with outcomes like decision quality, understanding and affective acceptance of decisions, while Jehn found that this was more true for teams performing more complex tasks. Affective conflict had an overall negative effect on outcomes measured across both studies.

After extended observation and interviews with collocated organization teams, Jehn (1997) expanded the task/relationship dichotomy by adding a third dimension – conflict about team processes. The study found process conflict to

be an overall negative force on team performance, similar to relationship conflict. The study also highlighted the importance of group norms around conflict in mediating these effects on outcome. The inclusion of process conflict in subsequent studies was somewhat inconsistent, in part due to difficulties establishing a clear distinction between task and procedural issues (de Wit, Greer, & Jehn, 2012). Recently, Behfar and colleagues have reexamined this dimension and shown its continued relevance to the conflict taxonomy, while distinguishing between different kinds of procedural issues (Behfar, Mannix, Peterson, & Trochim, 2011). Specifically, they found process conflict involved both a contribution dimension, such as the extent to which group members are living up to their assigned roles and tasks, as well as a logistical dimension concerning issues such as allocation of resources and time.

A significant portion of work on conflict in the last two decades was concerned with identifying the precise conditions under which the various types of conflict lead to positive or negative outcomes in a contingency model (Jehn & Bendersky, 2003; Korsgaard, Soyoung Jeong, Mahony, & Pitariu, 2008). In doing so, the studies frequently followed an input-process-output model (McGrath, 1984): structural factors such as team interdependence and task complexity formed the inputs, conflict was examined as one of the processes, in reference to team substantive outcomes like performance and quality, and affective, such as satisfaction. Literature Review 3 in Chapter 4 will focus in more depth on the relationship between conflict types, inputs and outcomes, thus it is not discussed here in detail.

A common criticism to this emerging taxonomy was the fact that the different conflict types were often found to co-occur or correlate with one another making it more difficult to tease out individual effects, especially in cross-sectional work (De Dreu & Weingart, 2003; Korsgaard et al., 2008; de Wit et al., 2012). A series of longitudinal studies provided significantly more clarity about the interaction of these different factors (Arazy et al., 2013; Greer, Jehn,

& Mannix, 2008; Jehn, 1997; Mannix & Jehn, 2003). The studies found that due to varying and often imperfect perceptions of team members about others' intentions, there may be a causal relationship between task and affective conflict (Jehn, 1997). Specifically, when task disagreements are misinterpreted as personal criticism, a task issue may transform into a relational one (Simons & Peterson, 2000). At the same time, some team members may treat task assignments as a personal reflection of their performance and character. Thus criticisms directed at their task competency, or the assignment of a task, may lead to hurt feelings and frustration (Huang, 2010; Jehn & Bendersky, 2003). Yang and Mossholder (2004) suggest that it is necessary to resolve task conflict during personal interactions, lest it turns into relationship conflict. Interestingly, theory consistently conceptualizes a link from task to affective conflict more often than the other way around, suggesting that affective conflict is a possible escalation mechanism.

Furthermore, process conflict has been found to exacerbate both task conflict, as well as the link between task and relational conflict. For instance, Arazy and colleagues (2013) found that task conflicts can gain a procedural dimension when discussions about what information to include in an article evolve into disagreements about interpretations of Wikipedia contribution rules. At the same time, Greer et al. (2008) have found that high process conflict levels early on increase the likelihood of increased levels of not only process, but also task and affective conflict in future. Finally, Martinez-Moreno and colleagues (2012) suggest that relationship conflict is prompted by the interaction between task and process conflict over time. In other words, when coupled with high levels of process conflict, task conflict is especially likely to trigger relationship conflict.

Taken together, the intragroup conflict taxonomy shows that conflict not only takes different forms, but that these forms are fluid, subject to interpretation based on personal perceptions and group norms, and evolve over time. Given

this evolving nature of conflict, it is relevant to consider conflict's role in group development as a whole. The next section will make a case for studying group processes and their evolution not just longitudinally over time, but also in real world (natural) teams that have a long-term orientation toward working together. A review of key group development theories will be presented with a particular focus on conflict, followed by empirical work that has examined either long-term oriented teams, group development, or both, in connection with conflict in collocated teams.

2.2.4 Team development over time

Time Orientation versus Group Change

It is important to distinguish between studying the effects of time in a group, and studying group change and development (Arrow et al., 2004). While time in team research is often treated as a methodological problem or a resource to be managed, group change, on the other hand, happens over time in specific patterns, and often follows non-linear dynamics. While it is possible to do longitudinal work on teams that do not have a long-term outlook, such as short-term student teams, these findings may differ from the nature of processes in teams that spend more time together, grow and evolve.

As Zaheer and colleagues (1999) point out, our observations of teams, measurement and analysis can happen across very different blocks of time compared to a team's actual life cycle. Furthermore, the validity interval of the inferences made may vary widely – teams with short-term orientations that expect to work together for only a finite amount of time can still vary widely in what this finite period is, from one experimental session, to one day, one project, one semester, and so on. Similarly, observing ongoing teams even for a relatively short period of time may yield somewhat different findings than observing short-term teams over several time periods.

For this reason, Saunders and Ahuja (2006) call on team research, and par-

ticularly, research on distributed groups, to distinguish between short-term oriented (temporary) and ongoing organizational teams based on the life span of their tasks. Specifically, short-term teams engage in one or a small number of concrete tasks that are finite (such as student teams producing a report over the course of a semester). Ongoing teams, on the other hand, engage in a multitude of tasks that may be repeated, in order to achieve recurring goals (such as a marketing team of an organization that has varying performance and outcome goals every quarter, but goes through repeated cycles of work related to each project). Crucially, ongoing teams have an expectation of future interaction in working together while short-term oriented teams do not (Axelrod & Hamilton, 1981). As a result, Saunders and Ahuja (2006) argue that ongoing teams have dramatically different team processes that impact outputs in contrasting ways.

Bona fide groups represent a kind of ongoing team (Putnam & Stohl, 1990). Bona fide groups are a collection of individuals with shared goals or aims that have 1) stable but permeable boundaries and 2) contextual interdependence not just internally but also externally. Stable but permeable boundaries do not only mean a fluctuation in team membership, although as team development literature shows, this is a key element in internally driven team change (Moreland & Levine, 1982). Permeable boundaries also involve communication with other teams, overlapping boundaries of some teams due to multiple team memberships, as well as team members having relationships in other contexts (be it a personal or work related context). Taken together, permeable boundaries shift the internal dynamics of the group through fluctuations such as new information, resources and perceptions.

At the same time, contextual interdependence is not only a set of variables that may impinge upon a group (Putnam & Stohl, 1990). Context is also embedded in the interactions of members with each other over time. Specifically, the probability of interactions with each other in other contexts, as well as prior history, influence group member behavior, even when the situations are unre-

lated, creating an expectation of future interaction. Real organizational virtual work groups, as well as online communities like Wikipedia and Free and Open Source Software projects represent good examples of bona fide groups because they have relatively stable but permeable membership boundaries, and a sense of context and history among team members that develops over time. The next sections will highlight how organizational teams change over time, and connect this with literature on collocated ongoing (or bona fide) groups and conflict.

Approaches to team development

Research that focuses on understanding how teams develop over time often takes either a sequential developmental approach, or a non-sequential task-based cycle approach (Mathieu & Rapp, 2009). The developmental approach believes that teams change according to a predictable and fairly stable pattern over time, while the task-based cycle approach instead focuses on understanding recurring patterns of interactions.

One common aspect of the developmental approach is the study of team life cycles, that is, prescribed sequences of activities that unfold in steps that a team necessarily has to take (Kuhn & Poole, 2000). Tuckman's (1965) forming, storming, norming and performing life cycle is one of the most commonly used frameworks that is still relevant to research today. Tuckman argues that all teams experience the same four steps in their team development. The forming stage is a period of uncertainty early in the life of a team as individual members get to know one another, establish roles, identify or clarify their goals. The storming phase is characterized by a phase of disagreement, which brings to the surface different interpretations of team goals, processes, or member roles. Through a successful resolution of these conflicts, the team develops their own rules, procedures and shared understanding (the norming phase). Having successfully negotiated the team norms, a group can then enter the performing phase where goal related activities are accomplished.

Thus Tuckman's life cycle assumes that teams have to perform a certain amount of activity not directly connected with the task at hand, such as the construction of common norms, before they can be successful. Tuckman's framework also implies that the teams have a fixed end point at some time when the goals are accomplished (Kuhn & Poole, 2000). Building on Tuckman's original work, Hill and Gruner (1973) found that teams do not necessary have to go through all the steps in sequence. Specifically, there is a carry over effect of member continuity and familiarity with each other from earlier teams or tasks that allows the team to skip some of the initial stages.

Later work studying team life cycles has found that groups may follow many paths, and no one best combination of stages exists for teams because this generally depends on task features (e.g. Poole & Roth, 1989). Specifically, work by Marks et al. (2001) shows that some activities, such as conflict or its resolution, are continually revisited by teams, and there may also be periods of largely unorganized activity in between task focused sprints.

On the other hand, task-based cycle theories focus on repeated processes, and the ongoing interplay of factors that influence future interactions. One of the most prominent examples of this approach is Gersick's (1988) punctuated equilibrium model, also still commonly used in literature today. Gersick argues that teams exist in a mostly stable structure that is punctuated by short bursts of radical change. The sources of change can be either internal or external, and result in a temporary period of instability until a new equilibrium is established. Gersick further suggests that this radical change occurs toward the midpoint of the team's life cycle or task. Interestingly, studies show that teams undergo a midpoint transition regardless of the total time taken in the project. For example Waller, Zellmer-Bruhn, and Giambatista (2002) find that both teams with stable and fixed deadlines, as well as teams with changing deadlines, experience a transition near the midpoint of the task cycle.

Worchel (1994) unpacks this cycle of transition into concrete stages, bridg-

ing the gap between the development and task-cycle theories. Worchel argues that when teams exist in a state of discontent among one or more team members, this reduces their identification with a team. However, a precipitating event may help to reaffirm the bonds between the team, leading to stronger identification and group productivity. A period of success may result in individuation as team members begin to take personal responsibility for their achievements, leading to a more competitive environment, and eventual decay. This stage is marked by a reduced sense of competition, identification and overall apathy, until the next precipitating event repeats this cycle.

Finally, Marks and colleagues (2001), in their temporally updated framework for team processes, suggest that teams experience repeated and varying cycles of action phases and transition phases. Action phases involve task-focused work towards collective goals, while the transition phase involves evaluation of those goals and activities, planning, and norm formation. Additionally, conflict can occur in both phases, either directed towards the goal and outcomes, or towards the transitional processes.

Taken together, despite their difference in focus, both the developmental and task-cycle approach agree that disequilibrium, disagreement and conflict are all crucial parts of the team development process. Furthermore, conflict and the formation of group norms appear to come hand in hand. For example, Raes, Kyndt, Decuyper, Van den Bossche, and Dochy (2015) examine team learning alongside Tuckman's life cycle stages. They find that the bulk of team learning happens only after teams pass the conflict stage. More importantly, the way in which a group communicates about issues during their resolution affects how they think about them, as well as team future interactions (Hirokawa & Rost, 1992). Kuhn and Poole (2000) find that ongoing teams are better equipped to handle task issues after working through conflict because roles and norms are established that enable future performance. Conflict can therefore provide a clear direction for a group and allow it to analyze its objectives. It can also as-

sist in analyzing a problem, negotiating goals, making estimations of possible consequences, the roles and relationships established, and set norms for structuring a task (Kuhn & Poole, 2000). In fact, Opp (1982) suggests that explicit or voluntary discussion is one way for teams to form group norms. It is therefore possible that differing opinions about appropriate group norms could also lead to conflict.

The following section summarizes some of the most recent work on conflict in collocated teams, paying particular attention to work that looks at issues of time or group development in combination with conflict.

Conflict, Time and Change in Collocated Teams

Research on organizational teams frequently conceptualizes team processes as actions arising from (often structural) inputs that lead to team outputs (Hackman, 1987; McGrath, 1984). This input-process-output (or IPO) framework has become the predominant way to understand team processes like conflict. The framework has many things in common with the structural approach to understanding conflict examined above. However, the original IPO framework had several limitations. Firstly, the framework implies a single and finite team life cycle with a linear progression from inputs to processes to outputs. However, classic literature on team development reviewed in the previous sections also identifies a cyclical and recursive dimension to team development. Thus outputs can become inputs for future team iterations, while processes like conflict can interact with both other team states like identification and trust, and directly influence team inputs in future iterations as well (Ilgen et al., 2005).

Second, as Marks et al. (2001) point out in their temporally updated process framework, not all mediating factors between team inputs and their outputs are necessarily team processes. Specifically, Marks and colleagues distinguish between team processes and team emergent states. A process is an interdependent act between members that converts inputs to outcomes through member activ-

ities focused on organizing taskwork. Emergent states, on the other hand, are dynamic properties of the team that vary as a function of the team's context, inputs, processes and outcomes. Therefore, while conflict is a team process that either supports or detracts from taskwork and the collective aims, it can also continually impact emergent states, or group psychological traits, such as group norms and shared mental models, identity, trust, and affect (Marks et al., 2001).

In an effort to acknowledge this view of teams as complex, dynamic and adaptive systems, research is moving toward an Input-Mediator-Output-Input (IMOI) model instead (Ilgen et al., 2005). The final "Input" in the IMOI model is reflective of a more cyclical system, while the M represents mediators that are both processes and emergent states that are able to influence each other. Examining conflict in teams from this perspective has lead to a number of interesting insights for research on collocated teams.

First, emergent states mediate the relationship between conflict and outcomes. Jehn, Greer, Levine, and Szulanski (2008) explicitly examined the impact of conflict dimensions (task, process and affective) on team emergent states (such as trust and identification), as well as their collective impact on team outcomes like performance and viability. They found that conflict had a negative impact on outcomes like vitality through reducing levels of positive emergent states. Therefore, conflict continually impacts the psychological climate of the team, which in turn, leads to changes in team performance.

Second, work by Greer and colleagues further shows that conflict not only affects emergent states; current disagreements may also influence the manifestation of other conflicts in the future (Greer et al., 2008). Specifically, the presence of unresolved procedural issues leads to higher levels of procedural issues in the future, as well as greater task and affective issues.

Third, conflict dimensions have different effects at different stages in the team life cycle. Integrating the conflict taxonomy with research on group development, Mannix and Jehn (2003) found that high performing teams experi-

enced greater task conflict at the midpoint of the group's life cycle, supporting the punctuated equilibrium model (Gersick, 1988). They also found that high performing teams experienced low overall levels of affective conflict throughout the team's life cycle, with steadily increasing process conflict towards the project deadline.

Therefore, across the life cycle of a team, different levels of various types of conflicts may be appropriate at each stage. Similarly, Tekleab, Quigley, and Tesluk (2009) examined the punctuated equilibrium model alongside conflict and its management in a longitudinal study of student teams. They found that managing conflict successfully at the midpoint was a primary mechanism to overcome early inertia in the team, develop cohesion around revised task strategies, and forge a stronger team identity.

In examining the relevance of conflict to different team stages, the above research has assumed teams to have a fixed life cycle with clearly defined start, middle and end points. However, real world teams are often more long-term oriented and evolving, spanning multiple projects with membership that fluctuates over time. Thus they may have multiple midpoints across different tasks, and the effect of conflict episodes can spill over into future project iterations. More research is needed to understand how conflict impacts team development in ongoing team settings.

Addressing this, Jehn and colleagues have recently outlined a series of theoretical propositions concerning how conflict may evolve in a team from dyadic to intragroup levels through a process of contagion (Jehn et al., 2013). They suggest that contagion occurs when coalitions begin to form among team members around opposing sides in a dyadic episode fuelled by emotional contagion and/or because the conflict episode threatens team outcomes. Furthermore, as more of the team gets involved in the episode, turnover possibilities increase, while outcomes and emergent states like performance, creativity, efficiency, and satisfaction reduce. Consequently, the resolution potential, and level of compet-

itive tactics employed, vary as the episode escalates.

Taken together, the above studies provide a number of important insights into the role of conflict in team development over time. Specifically, conflict and emergent states interact with one another to affect team outcomes and future inputs. At the same time, different kinds of conflict occur together and transform over time into other conflict types and new conflict episodes. Different kinds of conflicts also have different effects at different stages. However, despite the longitudinal design of this work, most of the above studies were conducted in a controlled setting using short-term oriented student teams. It would be useful to examine these dynamic effects in bona fide, or ongoing teams, as well.

Having mapped out the state of conflict research in collocated teams to-date, the next natural question is how has literature on virtual teamwork handled the subject? The following section will examine literature thus far on conflict in virtual teamwork, and argue that a gap exists in understanding the role and impact of processes in ongoing bona-fide groups. Drawing on research on collocated teams, the chapter will formulate a research aim for studying conflict in ongoing virtual team settings.

2.3 Virtual teams

Early research on virtual teamwork had largely focused on understanding the impact of mediated communication on effective team interactions. This arose from an assumption that computer-mediated communication is inferior to face-to-face interactions for coordination. The present section first reviews work in this paradigm of "CMC as lack" and relates it to empirical work on conflict in virtual teams. Following this, the chapter presents alternative perspectives on computer-mediated communication that argue virtual teams can approximate face-to-face interactions, especially given enough time. These perspectives will also be supported with a review of relevant empirical evidence form virtual team

research, with a particular focus on conflict. Finally, the section will conclude with an overview of research gaps in the space of conflict research in virtual teams, that is, a lack of research on ongoing real world teams, in particular with respect to the role of conflict in team development and formation of group norms.

2.3.1 CMC as "lack"

Early research on computer-mediated communication focused primarily on documenting the differences between virtual and face-to-face interactions, and their implications. Short and colleagues (1976) argued that computer-mediated communication provides a reduced degree of awareness about the psychological presence of others, called social presence. This makes communication more difficult, and thus if a high degree of relational interaction is required, a medium with higher social presence would be necessary to ensure successful communication. Similarly, the media richness theory proposed that communication media vary in their ability to change an interlocutors' understanding within a time interval (Daft & Lengel, 1984). Rich media are better able to overcome ambiguity due to varying frames of reference between communicators. Lean media, on the other hand, take more time to convey a similar level of understanding. Text based computer-mediated communication such as e-mail is considered fairly lean, while video conference calls are much richer, and face-to-face interactions provide the highest benchmark for richness. Thus, particularly during negotiation or discussion of competing ideas, richer media are preferred because they would be less equivocal and offer less opportunities for misinterpretation.

Together, these perspectives became known as the cues-filtered-out approach (Culnan & Markus, 1987). As the name suggests, according to this view, computer-mediated communication had a distinct lack: a lack of social presence making it less efficient and a lack of richness resulting in more ambiguity. Thus relationship maintenance via CMC was seen to be more challenging and

with interaction likely to be subject to greater misunderstandings.

This early work was instructive in understanding how structural features of virtual communication created practical differences from face-to-face interactions. In fact, early research on conflict in virtual organizations found that certain structural features of CMC sparked or intensified conflict. For example, the dispute-exacerbating model of e-mail (DEME; Friedman & Currall, 2003) argued that the asynchronous nature of CMC makes it more difficult to time actions and reactions, while the lack of contextual cues compromise mutual understanding. Relational cues like politeness and sarcasm are particularly difficult to transmit (Simons & Peterson, 2000). If messages are not well understood and reactions are ill-timed, this can exacerbate conflict (Jehn, 1997). Hinds and Bailey (2003) also argued that without a common understanding of an issue, conflict is harder to resolve.

Information exchange can lead to conflict in virtual settings both due to a lack of information exchanged, and information overload (Kankanhalli et al., 2007). For instance, silence from the other party after a communication exchange may be difficult for team members to interpret due to a lack of contextual information available – i.e. is the team member away, busy working on something else, or simply not putting in enough effort? (Cramton, 2001) Information gathering is also more difficult as it takes more time, and introduces the possibility of excluding team members from the communication loop (either accidentally by missing out a carbon copy in an e-mail, or maliciously) (Cramton, 2001; Griffith, Sawyer, & Neale, 2003). At the same time, attempting to stay up to date on too many different activities and teams may also lead to information overload, thus making it more difficult to distinguish important pieces of information and the possibility of missing out on important details (DeSanctis & Monge, 2006). Together, these misunderstandings and coordination difficulties can stimulate conflict.

The relative invisibility (and at times, anonymity) of conversation partners

can also lead to more deregulated behavior (Sproull & Kiesler, 1986). For example, Kiesler, Zubrow, Moses, and Geller (1985) compared interactions between dyads (two individuals) in computer-mediated and face-to-face settings and found that individuals were evaluated less favorably, and behaved in a more uninhibited manner in the CMC setting. They concluded that behavior online is less inhibited due to the anonymity afforded by computer-mediated communication (Sproull & Kiesler, 1986). As a result, perceived divergent interests are more likely to manifest into conflict due to highly deregulated behavior. Literature on virtual teams has highlighted this challenge of deregulation to distributed work and suggests therefore that conflict will be a largely destructive team force (Griffith et al., 2003).

Taken together, early computer-mediated communication theories shared a common assumption that virtual interactions experience a kind of "lack" when compared with face-to-face encounters. Thus CMC communication is more difficult than face-to-face, more rife with conflict, and therefore cannot be used exclusively to build relationships.

Empirical research in this tradition that investigated conflict often focused on three related areas: comparing virtual and collocated teams across various degrees of virtuality; investigating the effect of greater heterogeneity of distributed teams; and differences in perception of remote team members, such as attribution of intent.

For instance, Hinds and Mortensen (2005; 2001) focused on understanding the emergence of conflict in virtual teams as compared with collocated teams. They found that greater use of computer-mediated technologies (virtuality) increased task (Mortensen & Hinds, 2001) and affective conflict levels in teams (Hinds & Mortensen, 2005; Mortensen & Hinds, 2001). Lira and colleagues were interested to understand whether conflict had more negative outcomes in virtual teams (Lira, Ripoll, Peiró, & Orengo, 2008). They contrasted task and relationship conflict effects on team potency in both collocated and virtual teams

and found that task conflict and relationship conflict reduced virtual team potency (perceived team performance), but not the potency of collocated teams.

Interestingly, studies have consistently shown that both geographically distributed and collocated teams used a similar level of computer-mediated technologies for their interaction, prompting later research to reformulate virtuality as a continuum with several dimensions including use of CMC, geographical distribution, time zone distribution, cultural heterogeneity, organization and work practice diversity (Chudoba, Wynn, Lu, & Watson-Manheim, 2005; Kirkman, 2005). Shin (2005) connected these dimensions of virtuality with distinct and different sources of conflict. Shin argues that spatial and temporal dispersion would be connected with greater task conflict, role and responsibility ambiguity with process conflict. Cultural dispersion or heterogeneity will be associated with more disagreements due to cultural differences, and therefore interpersonal or relationship conflict, whereas organizational dispersion would be connected with weaker team identity, and lower team cohesiveness.

Supporting these propositions, Kankanhalli et al. (2007) found that cultural diversity is connected with both task and relationship conflict, while functional (work background) diversity is connected with more task conflict only. Similar to work on collocated teams, they also find that task conflict's impact on performance is connected with the level of task complexity, while affective conflict's impact on performance is greater when teams are more task interdependent. Furthermore Rutkowski, Saunders, Vogel, and van Genuchten (2007) found that teams with a high temporal disassociation (time zone distribution) who were also highly immersed in their task experienced more interpersonal conflict than teams in the same time-zone.

However Staples and Zhao (2006) found that while virtual teams are indeed more culturally heterogeneous than collocated groups, it is the heterogeneity rather than the virtual nature that is responsible for greater conflict. That is, while overall culturally heterogeneous teams experience greater conflict, there

is no difference in conflict levels between collocated and virtual teams. In a recent meta-analysis, Ortiz De Guinea, Webster, and Staples (2012) found that virtuality does have an impact on short-term distributed teams (working together less than a day, such as student teams), but not on longer-term groups (more than a day old).

Finally, research suggests that virtual groups may make different, and often erroneous, attributions about their remote team members. For example, Cramton (2001) finds that team members make misattributions of others behavior due to lack of mutual knowledge and shared context/situations. Specifically, issues like poor performance or lack of communication are attributed to remote team members negative disposition (such as tardiness) rather than ameliorating situational factors such as connection difficulties. Furthermore, Walther and Bazarova (2007) find that virtual team members are also more likely to attribute their own negative behavior such as poor performance to the behavior of others in distributed teams, as compared with both collocated and mixed groups. Both types of attribution biases suggest that task conflict is more likely to be interpreted as or transform into relationship conflict in distributed work. However, Martinez-Moreno et al. (2012) found that early task conflict predicted later relationship conflict only in collocated teams and teams that used virtual conferencing tools, but not text based computer-mediated communication tools. This may be because lean communication media also afford more time to construct a response, thus reducing possible misinterpretation. This may also be a result of transmitting less cues about annoyance, such as facial expression and tone of voice, that prevent issues escalating into affective conflict. These findings suggest reduced social cues may have differential, that is, both positive and negative, effects on team interactions.

Taken together, the above work suggests that while CMC tools present some challenges to virtual team interaction, there may also be advantages. The next section presents work that elucidates mechanisms through which CMC-driven

teams may function just as well as their face-to-face counterparts.

2.3.2 Beyond limitations of reduced social cues

One potential concern with the transmission of less social cues and a reduced sense of presence is the greater difficulty in establishing a sense of shared identity between team members that enables more cohesive teams. However, more recent CMC theories suggest that a strong sense of identification can be established in virtual teams as well, both relatively quickly and over time, with varying effects. This section will briefly review these theories, together with relevant empirical work that highlights their impact on conflict.

The Social Identity Model of Deindividuating Effects (SIDE) offers one such perspective (Lea & Spears, 1991; Reicher, Spears, & Postmes, 1995; Spears & Lea, 1994). SIDE argues that limited social information about participants in virtual environments can actually heighten identification, rather than reduce it, under certain conditions. While CMC does indeed filter out certain interpersonal communication cues that may reduce the richness of communication on an interpersonal level (such as others' facial expressions and nonverbal cues), category level cues can be communicated regardless of richness and bandwidth (such as location, nationality, age, gender, etc.) (Spears & Lea, 1992). The SIDE approach argues that the more easily communicated categorical identity cues become more salient under conditions of invisibility in text-based communication media. This encourages self-categorization behavior from team members by increasing awareness of an overarching group identity. Empirical work has found support for this proposition both in lab studies of short-term teams (Reicher et al., 1995), and by observing naturally occurring groups over time (Postmes et al., 2000).

The SIDE model suggests that the absence of individuating information about the team that would typically be present in face-to-face interactions, virtual teams may in fact be more cohesive, have a stronger level of group iden-

tification, and experience less conflict due to misunderstandings and interpersonal friction. Recent work has found support for this assertion. For example, Windeler, Robert, and Riemenschneider (2015) manipulated distributed team member profiles to heighten similarities and found this reduced the levels of task conflict within the team, while improving shared understanding and team effectiveness.

This quick self-categorization and identification may be a mixed bag for virtual teamwork because it may also lead to the formation of subgroups and faultlines (Lau & Murnighan, 2005). Polzer, Crisp, Jarvenpaa, and Kim (2006) suggest that when teams are distributed across different locations or time zones, this may lead to the formation of subgroups because the more immediately accessible local group identity would become more salient than the overall group identity. Similar effects may occur if team members have a heightened awareness of other characteristics, such as a split along functional backgrounds (for instance, users who prefer to use Vim versus Emacs, or software engineers versus non-technical staff), gender (if it is made visible in virtual profiles), or more arbitrary individuating categories. These faultlines may also be exacerbated in teams where more than one kind of characteristic is present in each subgroup. Furthermore, O'Leary and Mortensen (2010) found that in distributed teams, the effect of conflict on outcomes was particularly bad if there were two imbalanced subgroups (one subgroup being considerably larger than another) rather than equally sized groups or heterogeneous groups without geographic faultlines, because this creates uneven power dynamics.

Thatcher, Jehn, and Zanutto (2003) found that the relationship between faultlines and conflict was not linear – groups with strong faultlines (two distinct and even subgroups) and teams without subgroupings experienced more conflict than teams with moderate faultlines (several subgroups). This is because strong faultlines lead to the formation of factions, or coalitions, that reduce team identification in favor of identification with a subgroup, thus engender-

ing conflict between the two sides. However the presence of several different subgroups ensures that no single coalition dominates, thus reducing potential for inter-subgroup conflict. At the same time, teams with multiple subgroups are more likely to have members that belong to more than one subgroup. This creates more communication links among team members and a sense of history, especially as different combinations of subgroups come together on different tasks. Thus teams with more subgroups experience less conflict than teams that are completely dispersed.

Relational information can also accumulate gradually despite the relatively lower bandwidth afforded by computer-mediated communication, according to the Social Information Processing theory (SIP) (Walther & Burgoon, 1992). Thus given enough time, team members interacting via relatively leaner media have the potential to get to know each other well and approximate face to face interaction (Walther, 1996). The theory has received consistent empirical support showing that over time, distributed teams are not only able to approximate face to face interactions, they can also learn to adapt to their environments and perform effectively (e.g. Chidambaram, 1996; Walther, 1997; Walther, Anderson, & Park, 1994). For example, Chidambaram, Bostrom, and Wynne (1991) found that over time, distributed teams were not only able to adapt to the use of group decision support systems software, they also displayed more productive conflict management and better cohesion than collocated groups.

Bringing the faultlines and SIP perspectives together, recent work by Yilmaz and Pena (2014) found that while subgrouping did lead to biased information sharing and more conflict, subgrouping was affected by both early stage social categorization as well as interpersonal behavior over time. Furthermore, interpersonal behavior was able to override the short-term categorization effects, such that initial in-groups or disagreements were moderated by the extent to which team members behaved in a positive or negative manner toward one another over time.

Taken together, these perspectives on relational communication in virtual teams suggest the importance of considering long-term effects when studying processes like conflict. Long-term virtual teams would be more likely to adapt to their environment and experience a more positive experience even in the face of conflict, compared with short-term teams that rely on categorization for group identity.

However, many studies on virtual team work, and in particular those that examine conflict, employ short-term or student teams in their research designs. Table 2 (Appendix .2) summarizes studies concerning virtual work performed thus far, and classifies them by their methodology, study subjects, and the tenure of teams involved. The table shows that with a few notable exceptions, studies employ largely student or other short-term teams that have no expectations of working together in the future beyond the study task assigned. This is true even in longitudinal research designs. While this has been a critical methodological tool to establishing a point of comparison in experimental studies involving both collocated and virtual teams, there are a number of interesting insights to be gained from studying virtual team processes in ongoing groups (Gilson et al., 2014; Martins et al., 2004). For instance, Martins and Schilpzand (2011) suggest that not enough work has been done to understand the nature of ongoing virtual teams, such as occurrence and effects of conflict, impact on long term commitment and team vitality, or how norms are developed. The next section will review studies on ongoing virtual teams and outline a concrete research gap and agenda.

2.3.3 Conflict in long-term oriented virtual teams

Some interesting patterns can be observed when examining the intragroup conflict taxonomy in virtual teams with a longer orientation toward working together. Hinds and Mortensen conducted a series of studies on real organizational teams. In the first study, teams had an average tenure of 15 months (Mortensen

& Hinds, 2001). They found that while distributed teams experienced more task conflict than collocated teams, the same was not true of affective conflict. Furthermore, distributed teams experienced greater benefits of shared identification in reducing task conflict levels than collocated teams. In a subsequent study, though team tenure was not reported, the participating employee tenure within the organization was an average of 9 years suggesting team members would have a high sense of history and expectation of future interaction (Hinds & Mortensen, 2005). They found both task and affective conflict levels to be higher overall in distributed teams, but that teams with shared identification moderated the negative effects of affective conflict and shared context of task conflict. Greater levels of spontaneous communication between team members improved both team shared context and identification, suggesting a support for the social information processing theory.

Similarly, Han and Harms (2010) found, among real world distributed teams in both Fortune 500 companies and a healthcare organization, that identification lowered both task and affective conflict occurrence. Furthermore, Jong, Schalk, and Curseu (2008) studied very diverse combinations of real world virtual teams from project and work teams, to management, study teams, spots and music groups. When controlling for team type and tenure, they found that highly virtual teams experienced more positive effects of task conflict on performance, than teams with low virtuality. However, all teams experienced negative effects of process conflict. Finally, similar to ongoing collocated teams, conflict does not remain static and immutable in ongoing distributed teams. Arazy and colleagues (2011, 2013) found that task conflict in combination with high cognitive diversity improved the quality of Wikipedia article output. However, when the conflict evolved into affective or process conflict, this dramatically reduced team output quality.

Taken together, these findings suggest that over time, ongoing virtual teams are able to establish strong bonds among team members and this has beneficial

outcomes for reducing both team task and affective conflict levels. At the same time, procedural issues remain a negative team force across different levels of virtuality and organizational structure. However, while ongoing virtual teams benefit from having higher levels of task conflict than teams with low virtuality, they are also more susceptible to task conflict transforming into less productive conflict over time. Thus, similar to collocated groups, ongoing virtual teams appear to require the negotiation of a fine balance between team cohesion, preventing groupthink through task conflict, and managing conflict escalation. However, we still know relatively little about how virtual groups continually negotiate these tensions over time.

Adaptive structuration theory offers some insights into how long-term virtual groups can evolve not just in terms of their interpersonal relationships but also in their technical and social processes (DeSanctis & Poole, 1994). The theory proposes that technology is not just a facilitator of virtual team interaction; it may also be responsible for molding the subsequent structure of the team. At the same time, teams may not only adapt to the technology used, they may also adapt their communication tools themselves to satisfy emerging social conventions in a dynamic and reciprocal process. Technology use is therefore socially constructed by appropriating existing tools in new ways, as well as building new tools; that is through both production and reproduction. How does this relate to conflict in ongoing virtual teams? There is growing evidence that conflict as a social process is a driver of both personal, social and structural group changes in virtual teams.

First, conflict can be a driver of personal learning and growth among team members. Bosch-sijtsema (2007) examined two related real world virtual work teams in a qualitative study. The first team tenure was 3 years, with the second 1 year. Furthermore, well-performing team members from the first team had an expectation of being selected for the second team upon completion of the initial project, suggesting a high expectation of future interaction. The authors found

that team members who experienced conflict due to a mismatch of expectations about working in the first team were more likely to depart. However, team members who detected the mismatches and learned from them were more likely to stay on and join the second team.

Furthermore, Campbell, Fletcher, and Greenhill (2009) found that conflict is crucial for establishing social order within ongoing virtual teams, such as online financial communities. Campbell and colleagues applied an ethnographic approach to the study of the community's interactions, and found conflict to be an important ritual that aids in defining and maintaining the fluid social roles found in online environments. In other words, conflict provides a set of common principles for understanding social interaction within the online community, and allows team members to explore and evolve various roles and identities. The authors further found that conflict between positions of power can help to align the values and ideals of an online community.

In addition to this, Ayoko and colleagues found that conflict can stimulate the development of group norms among short-term student virtual teams (Ayoko, Konrad, & Boyle, 2012). The study examined the classic team development stages proposed by Tuckman (1965) reviewed above: forming, storming, norming and performing. They found that groups who were able to move beyond the storming stage and develop group norms more quickly performed significantly better. Furthermore, the study traced how emotional conflicts at the forming stage had a relatively less strong effect on team performance than during the subsequent storming stage. Initial affective disagreements were somewhat expected, but continued emotional conflicts had a negative influence on the team over time.

Conflict can affect not only the social, but also the technical structure of the team. Stark, Bierly†, and R. Harper (2014) observed the effects of conflict in the decision of collocated short-term student teams to use more virtual communication over time. They found that greater levels of relationship conflict

experienced by the team were associated with teams choosing to communicate more virtually in the future. Greater procedural conflict was also associated with decisions to communicate more virtually, but only when the teams also experienced high levels of cooperation.

Taken together, gaps remain in our understanding of the role of conflict on ongoing virtual team development. Some prior work on ongoing virtual teams examines conflict as a monolithic entity, and either do not specifically define their conceptualization of conflict, or do not contrast their conceptualizations with the dominant conflict paradigms in organizational research. Work that does examine different types of conflict in ongoing virtual teams does not consider how conflict may differ due to expectations of future interaction. Finally, research that examines the role of conflict in impacting team social and technical structures continues to look at short-term student distributed teams.

Based on research on collocated teams, we know conflict is likely to transform, it is subject to team member interpretations based on their own perceptions and group norms, and has an impact not only on outcomes but also team structures and emergent states. Thus it is important to understand how conflict is interpreted by team members in ongoing virtual contexts, whether this differs from short-term teams, and how conflict impacts virtual team outcomes, future interactions and team structure over time.

Thus the present work aims to address these research gaps with the following research questions:

How does conflict as a process differ in dynamic ongoing virtual groups like FOSS projects compared to conflict in short-term virtual teams in traditional organizations?

How does conflict impact structural change in dynamic and ongoing virtual groups like FOSS projects?

Chapter 3

Study 1 - Interviews and Participant Observation

This chapter [1] presents an initial explorative study to understand conflict manifestation in Free and Open Source Software (FOSS) development teams as examples of ongoing and voluntary virtual teams. It takes the first step towards understanding conflict in FOSS teams in a systematic way. The study contrasts the emergence of conflict in FOSS with conflict theory in traditional organizations, through a grounded exploration of the FOSS community and interviews with diverse community representatives. The findings highlight both the similarities of conflict in ongoing voluntary virtual teams, such as FOSS teams, to our existing understanding, as well as notable differences that have important implications both to the study of virtual teams as well as management practice. At the same time, the chapter explores factors from participant accounts and observation that both stimulate and help to control and reduce conflict in ongoing voluntary virtual teams. These factors will underpin hypothesis development and design of the second study reported in Chapters 4 and 5.

[1] Material from Chapter 3 was published in Filippova and Cho (2015)

3.1 Study scope and specific research questions

The previous chapters have presented an overview of literature on both FOSS teams as well as more broadly on virtual work, interaction among virtual teams, and their experiences of conflict. The review has highlighted that, when taken together, the literature shows a number of gaps the present study is designed to fill. Firstly, while there exist a handful of studies describing conflict as a process in FOSS teams, this work is largely descriptive, focused only on a small number of very successful FOSS projects and does not differentiate between different dimensions of conflict (Elliott & Scacchi, 2003; Jensen & Scacchi, 2005; Wang et al., 2015). Thus the present chapter explores conflict in FOSS across variety of projects, recording perspectives from members of different backgrounds and experiences to build a model of conflict that is relevant and consistent.

At the same time, while a much more substantial body of work exists on conflict in virtual teams, these studies focus on conflict in largely short-term teams with fixed membership structures, and do not consider whether the expectation of future interaction may impact the way conflict manifests in ongoing teams, in particular those with voluntary membership and a greater level of self-determination. Thus the present chapter also takes the opportunity to explore differences in conflict experience for ongoing teams, and its role in impacting team structures and future inputs.

Finally, the present study takes a first look at what team structures may give rise to, or reduce, conflict manifestation, because FOSS research on conflict has not yet examined this relationship in great detail. Thus this first explorative study addresses the following research questions:

RQ1: How do FOSS teams experience conflict? Do ongoing virtual teams such as FOSS development teams experience conflict differently from short-term virtual teams? If so, in what way?

RQ2: How does conflict influence future team structures and inputs?

RQ3: What factors contribute to emergence and management of conflict?

3.2 Method

To better understand the ways in which conflict differs in FOSS development teams, and the way the process is interpreted and understood by participants themselves, the present work first begins with a grounded approach. This is because a level of immersion into the community is necessary to allow interpretation of conflict as a process and important variables emerging from and leading up to conflict from the point of view of FOSS developers. At the same time, while not starting from a blank slate, a level of detachment from theory during data collection was viewed to be essential in uncovering the extent to which theory on short-term virtual teams applied to the context of conflict in ongoing teams like FOSS, as well as the gaps that remain.

3.2.1 Epistemology

There exist a number of epistemological approaches to the practice of grounded theory. While it is not the focus of this work to engage deeply with this ongoing debate, the distinction must be mentioned in order to clarify the epistemological stance of this study and its consequent choice of application of grounded theory methods.

Grounded theory was first proposed by both Glaser and Strauss (1967) however, their contrasting perspectives on epistemology have led to a divergence of approaches. Because the different perspectives are derived from the same initial work, they share the following characteristics: 1) simultaneous data collection and analysis that iteratively shape the research process and ongoing theory building; 2) addressing research questions through data-driven emerging codes and categories, rather than preconceived hypotheses; 3) a constant comparison of categories relative to each other and emerging theory as a whole; and 4) theoretical sampling that continually selects participants based on the needs of the developing theory rather than a pre-set participant list (Charmaz, 2003).

The different approaches to grounded theory diverge primarily on the role of the researcher, and the extent to which she can locate an absolute truth within the subject and community of interest. As a result, their specific applications and tools differ. The Glaserian (1978) approach is concerned with the emergence of a theory that is a more accurate and parsimonious reflection of participant experience because it is independent as far as possible from the influence of researchers' pre-conceptions. However, critics of Glaser's approach question whether it is possible to achieve this level of detachment from theory. On the other hand, the Strauss and Corbin (1990) approach acknowledges that the researcher is engaged in a necessarily interpretive role, thus their subject-domain knowledge and experience are an initial shaping point to the emerging theory. However, the Straussian approach is criticized for its complex layer of processes aimed at emerging theory verification that may steer a researcher away from the data (Heath & Cowley, 2004).

Given their respective limitations, the present study takes a middle ground between the two approaches. Following the work of Charmaz (2003), who reframes grounded theory in a more constructivist framework, the present work takes into account not just the researcher's active role in shaping the emerging theory from data, but also the broader social, historical and situational constraints of the participants. The Charmaz approach has a few notable distinctions (Breckenridge, Jones, Elliott, & Nicol, 2012). Firstly, it strives to incorporate multiple perspectives of shared experiences, thus highlighting different interpretations among groups of participants on the same issue. Secondly, meaning is co-constructed by the researcher and participants. That is, while Charmaz' approach draws on the researcher's background, it does not exclude the researcher's participation in the community or situation being studied. Instead, it encourages reflecting on the emergent meaning with the participants and the academic community. Thus it is Charmaz' epistemological approach to grounded theory that is utilized by the present study, while continuing to apply

the traditional tools of grounded theory described above. The following section elaborates on the specifics of the data collection and analysis.

3.2.2 Data Collection

Data collection began in November 2013 using a combination of participant observation and in-depth interviews. This researcher began by attending local Meetup events in Singapore that focused on Free and/or Open Source Software based technologies with an intention to understand and observe the current debates, trends and critical issues being discussed. Such observation served as background information for the study, was recorded in a series of journals, and supplemented the data analysis of in-depth interviews by providing context about the community and technology. At the same time, attending Meetups afforded the advantage of personal introductions and significantly better participation rates for in-depth interviews than remote interview requests.

Singapore as a geographical starting point for the research offered a number of additional advantages. Singapore not only has its own thriving technology community, it is also the frequent recipient of visitors from both the Americas and Europe, as well as the rest of Asia and Australia, offering a wide mix of backgrounds and experiences. Thus, while the research initiated in Singapore, through connections made in the local technology community, attending events and referrals, this researcher was able to interact with participants from a variety of countries aside from Singapore, namely: Australia, Belgium, Canada, China, Germany, India, Japan, Malaysia, the Netherlands, Sri Lanka, Taiwan, and the USA.

3.2.3 Immersion

This researcher aimed to immerse herself in the FOSS community to better understand developer perspectives on conflict. At all times this researcher presented oneself transparently as a doctoral candidate studying the Free and Open

Source Software community. This researcher explored community connections in person at local technology events such as FOSS user groups, programming language meetups, as well as by visiting hacker- and maker-spaces. Furthermore, this researcher attended community events that were likely to attract international participants, such as regional conferences connected with open programming languages and FOSS tools. This researcher also attended workshops covering FOSS technologies. Over time, due to the relationships forged and time spent in the community, this researcher was invited by community members to participate more actively by helping to organize Meetup events, giving talks and assisting with conference organization.

3.2.4 Theoretical sampling

Interviews were conducted in stages across over a year while initial coding and the process of constant comparison produced a preliminary understanding that drove theoretical sampling of further participants. Thus participants were selected for their specific backgrounds and experience to close gaps of understanding across aspects of theory development. For example, when it became apparent that leadership was an important factor in the way conflict is experienced by FOSS teams, an effort was made to interview more core contributors and team leads on their experiences of conflict.

3.2.5 Interviewee demographics

In total, in-depth interviews were conducted with 22 participants, of whom 4 were women. Participants were a mixture of developers at different levels of experience and embededdness in the social network of the project: committers (both regular, experienced and very new), maintainers, core contributors, project founders (of both simple and complex projects) and community ad-

visors.[2] Projects included libraries and open standards: BitcoinJS, DMARC, libXML, PrawnPDF; languages, tools and frameworks: Git, GeoTools, Grok, Java, Morepath, Node.JS, Polyglot, Python, Ruby, Ruby on Rails, QuiverJS and Zope; and end-user applications: Debian, Mozilla Firefox and Thunderbird, Plone, ScrollBack, Ubuntu and WeBuild.SG.

Table 3.1 below highlights the distribution of projects sampled by type of leadership as well as project size. Small refers to projects that have 2 to 20 developers, while large refers to projects with 20 or more developers (sometimes several hundred). A "benevolent dictatorship" refers to a FOSS project with a strong leader figure who can make unilateral decisions without consulting other community members. Highly distributed decision-making typically involves decision making my rough consensus of a majority of developers, while projects run by foundations or committee have a more differentiated and formal hierarchy.

Though varying in size, leadership structure and stage of development, all the sampled projects are active and at least one year old, with a majority spanning several years. They therefore represent examples of ongoing virtual groups that have had a chance to evolve.

[2]Project founders refer to individuals who first wrote the piece of software that grew into the project studied. They are often initial project leaders, but may step down and allow the project to be run by someone else. Core contributors refer to a small group of individuals who contribute the most code to a project. Frequently these contributions span across different areas of the project. They are usually, though not always, recognized with an explicit title of "core contributor" by the other project members and broader community. Maintainers are developers who have responsibility over developing one specific area of the project, integrating the contributions of others and ensuring they are in line with project goals. Committers are members without an official title who have submitted one or more changes to the code base that has been accepted by a project. They are often recognized as ordinary contributors in project documentation. Finally, community advisors refer to individuals who may or may not directly contribute code, but have a high enough social status to influence decision making in a project, such as by giving advice.

Table 3.1: Projects involved in Study 1 by type of organization and size

	Benevolent Dictatorships	**Highly Distributed Decision-making**	**Foundations or Committee**
Small	libXML, polyglot, quiverJS	PrawnPDF, BitCoinJS, WeBuild.SG	Thunderbird Calendar, Scrollback
Large	Ubuntu, Ruby, Rails, Git, Python	Debian, GeoTools	Firefox, NodeJS, Zope, Plone

3.2.6 Interview Procedure

Interview questions were semi-structured, thus the order and nature varied depending on the progression of the interview. Questions covered a variety of areas aside from conflict directly, to avoid priming interviewees about conflict. To help participants feel more at ease, they were first asked to discuss their experiences in contributing to FOSS more generally, such as their motivations to join, and how decisions were made in the projects they actively participated in. Then the interview moved on to more sensitive questions, such as memorable instances of disagreement with other project members. Participants were prompted to recall the event sequentially – i.e. how did the disagreement emerge, how did it progress and what was the outcome (if any). Additionally, participants were not limited to discussing recent or ongoing conflict episodes - conflicts across different aspects of the project lifespan were discussed.

Participants also discussed conflicts that they have observed, but may not have been directly involved with. Interviews lasted approximately one hour and were transcribed for further analysis. Participants were encouraged to share additional materials after the interviews, such as news articles related to the examples discussed. These materials supported the analysis process in either providing more context or, because project communication of many projects is freely accessible online, direct evidence of experiences mentioned in interviews.

Informed consent was sought for all in-depth interviews conducted, and only information with explicit consent from participants is quoted in this work. All interviews were confidential, to allow participants more comfort in sharing sen-

sitive experiences about conflict within projects they are active in.

3.2.7 Analysis

Analysis was conducted throughout the data collection process, rather than at the end, in line with the grounded theory approach. The grounded theory approach distinguishes between initial codes, namely substantive codes that are closer to the transcript, and intermediate or higher order codes, namely theoretical codes, that bind the substantive codes in larger theoretical relationships (Glaser & Strauss, 1967).

Thus, transcripts were first open coded to identify substantive themes. Coding was performed on small sections of text, one or two sentences at a time, using primarily phrasing or summaries of the text. Conflict episodes were identified in the transcripts based on the definition derived in Chapter 2, that is the perception that other team members were blocking or in some way frustrating the goals of contributors. These may be based on personal experience of the participants, or an interpretation by participants that an event they have observed represents conflict to them. Other themes were also coded for their potential future relevance in influencing conflict constructs.

In the intermediate coding stage, codes were then combined based on similarity into more abstract theoretical codes that also reflected on the interrelationship between different code categories. These categories were revised and adjusted in a process of constant comparison as more data was collected through subsequent interviews. Potential relationships between the categories and their significance to theory development were recorded as memos in a separate journal. Memoing allowed to draw further on community observations and other examples supplied by interviewees to get a sense of history and trajectory of the conflict episodes.

The process of data gathering through interviews, archival look-up and participant observation continued until no new substantive codes were emerging

and new data did not change the broader theoretical codes, thus reaching theoretical saturation. It is only upon the completion of this process that comparison to existing conflict theory was made.

Table 3.2 illustrates the common codes used, and their final broader theoretical structure. Specifically, the first column represents initial substantive codes used, while the second column represents their overall theoretical connection to the issue of conflict in FOSS teams. The third column distinguishes between codes that represent conflict types and the factors that may account for conflict emergence or variation (inputs). The conflict categories and input categories are elaborated on separately and in turn, in the Results section 3.3 below.

Table 3.2: Substantive and theoretical codes based on grounded theory approach

Unique Substantive (Initial) Codes	**Theoretical Codes**	**General Category**
Communication delays No response from those involved in project Limitations of communication tools Commenting for the sake of commenting Hard to be heard among noise (information volume) Lack of paper trail/documentation	Communication medium	**Inputs**
Geographical dispersion	Team dispersion	
Interdependence between developers Software modularity/interdependence	Interdependence	
Trust in long-time (senior) contributors Leaders involved in resolving issues Differences in project governance	Leadership	
Personal Disagreements	Personal/emotional issues	**Types of Conflict**
Rigid group norms and groupthink Conflicting group norms from different projects Different paradigms, ideologies or mental models Unclear group norms	Group norms	
Idea, feature or roadmap suggestions Conflicting software versions or dependencies Implementation variations	Task ideas and variations	
Sidestepping process Coding style Giving credit (Unwritten) Rules Differing expectations about how to make contributions	Procedures	

Because coding and interpretation in the context of this study is very closely linked to being immersed in the community, employing multiple coders was impractical as other potential coders did not have the ability to be in the field together with the principal investigator of this project. Thus emerging codes and interpretations were validated with interview participants and other community members on an ongoing basis. Verifying codes with participants is an additional step that requires significant participant time. To avoid belabouring participants, verification varied from formal follow ups to clarify interpretation, where it was necessary and appropriate, to less structured observations of how other community members talk about a particular or similar issue, followed by probing questions during community events to understand how community members think about the issue.

Furthermore, conflict accounts were based on individual participant recall and subjective interpretation, and teams were often represented by one participant in the study due to the need to sample a wide variety of developers. To address this, efforts were made to triangulate the present study findings by locating specific examples, where possible, in the digital work traces of projects involved, such as mailing list archives, GitHub comments, blogs and so on. Facilitating this, participants often sent links to conflict episodes they had discussed in the interview, or that they have come across but did not mention in the interview. These were also coded and informed the emergence of both substantive and theoretical codes.

Therefore, in the present study, the role of participant observation and archival data was to reify interview interpretations, establish code validity, triangulate the findings and provide additional context to the examples raised in interviews.

3.3 Results

Overall, the findings show some overlap with our existing understanding of conflict in short-term virtual teams and underscore the continual relevance of Jehn's (1995, 1997) conflict taxonomy for ongoing teams. However, findings also show that there are several notable elements of conflict in ongoing teams that are distinct from short-term teams. Specifically, there is evidence of an additional conflict dimension not yet included in the taxonomy – conflict concerning the norms and values of the group, or normative conflict. Normative conflict is particularly relevant for ongoing teams, as it has long-term effects on future team structure by encouraging the formation, evaluation and crystallization of group norms, as well as the possibility of creating social factions. Furthermore, the study observes conflicts evolving from one type to another over time. Finally, the section concludes with several factors that stimulate or reduce conflict levels in FOSS teams.

3.3.1 Task Conflict

In short-term face-to-face and virtual teams, task conflict involves disagreements about the task that needs to be performed. In the case of Free and Open Source Software development, the broader task is the development of a software application, collection, library, or standard. Thus task conflict in FOSS teams is commonly concerned with the project road-map, implementation or dependencies. Task conflict is also by far the most commonly occurring type of conflict described by participants, in part because many communities actively stimulate it as part of the code review, or request for comment procedures.

Road-map

A project road-map is a defined and (usually) agreed upon plan for developing a set of features, improvements or enhancements to the existing project code base.

One common source of task conflict in FOSS teams involves features submitted by contributors that are not part of the formal road-map, or a more unofficial idea the project lead may have about the project direction.

"Pull requests[3], features-wise, are harder [to get accepted], unless you follow the road-map" (P9). Sometimes if a feature proposed is popular, or users want to see this feature included, they will put social pressure on maintainers to accept the feature through mailing lists or comments on the issue tracker page (P9). This can create a community wide conflict because maintainers are reluctant to accept features that are not part of their plan for the project (P10).

P14 recalls that a number of feature request conflicts have occurred between themselves, as a lead developer of a chat client, and the user base that suggest feature improvements but do not put in effort to put the suggestions into code. Due to limited resources, a lot of feature suggestions are ignored or rejected, unless a user is willing to put in effort to develop it. In Drupal this is an unwritten rule:

> *"Some people had their own cases which they would like to see addressed, but the core maintainers kinda had the attitude that unless you could contribute code for that, you don't deserve any attention" (P1)*

Thus feature requests and contributions outside of the road-map may be ignored, resulting in a latent task conflict that manifests when voices within the community try to drum up support for their idea.

[3]A "pull request" is a commonly used term in software development to denote a change to the code that has been submitted for review to a developer in charge of the relevant section of the software. Pull requests may or may not be accepted based on the information provided together with the change, how large the change is (smaller is usually better, but not too small), and any associated discussion (often publicly visible). Though the terminology originated from git, a version control system underpinning the GitHub platform, the term may also be used by developers more generally to denote requests for change made using other systems, similar to "Xerox" being eponymous for making a photocopy.

Implementation

One of the most common sources of task conflict discussed by participants was conflict surrounding different implementations (P2, P7, P6, P9, P11). An implementation is the realization of a technical specification or algorithm through programming and deployment. Crucially, many possible implementations may exist for a given specification or standard, "and [because] it can be done in a number of different ways... so that's where the conflict happens" (P7).

Projects often have a preference for implementation, thus conflict may emerge when a developer stands firm on an implementation that contradicts the idea team leaders or maintainers may have. For instance, P2 recalls a developer whose implementation was rejected by Git maintainers, yet he persisted, "sending it in again a few weeks or months without any changes". This task conflict continued unresolved because the contributor was unable to compromise on the implementation with the Git maintainers.

Implementation conflicts can also emerge because developers have "different truths" concerning the right implementation (P6). Developers are often engaged in several projects, and come to the community with experience in different paradigms of software development. These may clash with the way things are already done in the project, for instance "going against the grain" and insisting on an object oriented approach in a community that emphasizes functional programming (P13). These discussions can be particularly long and fruitless as a community is unlikely to move outside of its dominant paradigm (P13).

However, conflicts about implementation are also "good in open source" because they lead to better outcomes, such as more efficient code (P7). For instance, when P7 proposed an implementation to solve a particular issue in Thunderbird Calendar, they ended up modifying their approach following criticism from senior developers:

> *"The people who were supervising me, they sent some private kind of code. And then I tried that code, I kind of changed my implemen-*

tation in the end... because I thought that the other implementation was actually better." (P7)

On the other hand, conflict about implementation often involves extended discussion to explain and defend your implementation against other ideas proposed by the community. Such discussions require efficient text based communication because "making a case to someone requires some sense of which bit of your worldview they do and don't share" (P6). Conflict can arise from an inability to efficiently express this message:

"A lot of conflict happen because of the verbal communication that you have... you've got to exactly specify what you exactly mean by when you say this is not a good solution." (P7)

Furthermore, the emphasis on the "art of (verbal) communication" (P10) may turn developers away from participation if they think they are not good writers, or are not native speakers of the dominant project language (P10).

Competing dependencies

Finally, because FOSS projects exist in a larger ecosystem and frequently depend on other projects for substantial functionality, debates about which dependencies to use are another prominent source of task conflict. For example, P15 recalls having to negotiate different contributors' suggestions for libraries to include in the project:

"So this person jumped in, and be like, 'Hey, I have a library. Use mine!' And then the other person, who has commit rights, and he jumped in, he was like, 'I also have a library'" (P15)

Such disagreements occur at an intersection of competing priorities and agendas, as contributors may prefer to include their own libraries to enhance

their reputation in the community or due to familiarity with the code base. Because FOSS projects depend on the efforts of volunteers, the maintainer is challenged to weigh these suggestions against other approaches and come up with a solution that satisfies both the volunteers and the project aim:

> *"[...] and then there's also a big number library, which is written by someone else, which is more popular than both of their libraries combined, so I'm like, "How about, I don't know, let's use the standard?" (P15)*

Negotiating conflicting opinions on project dependencies is difficult not only because choosing a side means rejecting the suggestion of some team members. Software dependencies also have their own teams behind them. Thus, choosing to include an outside project also means choosing to depend on the project's community for parts of your own project function.

Code Review and Request for Comment

Task conflict is one of the most commonly occurring conflict types reported in part because many communities actively stimulate it through code review and request for comment procedures designed to initiate discussion. P7 explains how a Mozilla foundation project implements review as part of the contribution process: first a feature implementation is proposed, then, if it receives sufficient support, the pull request goes to code review. The code is either accepted or more discussion occurs if someone has a different idea and suggests changes. For P7, having submitted a feature implementation proposal through the Google Summer of Code, this was surprising:

> *"I thought I'm accepted, so I thought I'd implement it, but that's not the case. [It] still goes through a hierarchical process of criticisms, change, criticisms" (P7).*

This hierarchical process is designed to stimulate creative diversity through task conflict. Collaborate software platforms such as GitHub have built in features that assume a code review stage for submitted contributions, and frequently include a discussions page for disagreements. Other issue trackers allow the assignment of a particular bug or submission to a developer for review, and in absence of discussion facilities, conversations take place in public on mailing lists.

The Request for Comment (or RFC) is a more sophisticated procedure for stimulating task conflict. RFCs were first used by the Internet Engineering Task Force (IETF, 2009) at the end of the 1960s to document and describe implementations of features or other solutions. RFC memos are frequently submitted for peer review, or to notify of new developments and invite comments from the public in the event there may be strong objections. As such, RFCs may be more complex than a regular pull request submitted for review. Furthermore, unlike pull requests that are submitted after the fact, RFCs are frequently proposed before an implementation.

The GeoTools project makes use of RFCs to coordinate work among multiple stakeholders and stimulate discussion of potential problems before work begins, to avoid rejecting contributions (P11). The GeoTools team has evolved the RFC procedure following instances of coordination problems that resulted in task conflicts, such as"watching like two or three groups all do the same work, and then have a terrible time when it comes time to try and merge their work back together" (P11). For GeoTools, an RFC system is therefore crucial to coordinate taskwork among diverse audiences.

Thus, taking together the above findings, task conflict appears to have many faces in FOSS development, including disagreements about the project roadmap or direction, implementation of a specific task, or the relevant dependency. At the same time task conflict is actively encouraged through both technical and social structures as it stimulates the generation of opinions and leads to more

productive outcomes.

3.3.2 Process Conflict

In short-term face-to-face and virtual teams, process conflict is traditionally defined as a disagreement about how to accomplish a given task, such as duty or resource allocation (Jehn, 1997). Behfar and colleagues (2011) separate process conflict into two distinct sub-dimensions: logistics, such as the amount of time to spend on a task and how task should be assigned, and contribution, such as whether group members are living up to their expected assignments. FOSS teams are largely voluntary in nature, and therefore disagreements about task assignment and responsibility (contribution) are somewhat less frequent because of the self-selecting way in which tasks are chosen. Logistics, however, are a prominent source of procedural conflict in FOSS teams, and include issues such as the style of writing code, the need and nature of testing and copying/distribution procedures. Logistics are not concerned with how to accomplish one specific task (this is a common source of overlap with task conflict). Rather, logistics involve how tasks in general are performed by the group of developers in question, and therefore process conflict is more broadly concerned with how software should be written.

P2 suggests that style is one of the first reasons for criticizing or rejecting a contribution in the Git project.

> *"Most of us think style doesn't matter because [the] computer doesn't care about white space and stuff, but [we have] no choice but to just follow" what the team lead says (P2).*

Style usually involves conventions concerning formatting, such as indentations, or more broadly about the structure of patches, for instance, preference for atomic patches that are small edits rather than a big chunk of code. Many larger projects (such as Linux and Ruby on Rails) formalize their contribution policies

into a published style guide. Style expectations can be very detailed, such as requiring contributions to be sent inline via e-mail, rather than as an attachment. This is because most contributors use command line mail readers, and attachments would interrupt their usual flow (P2). P2 suggests that while the rules appear to be arbitrary, they serve a critical function in allowing collaboration in a more standardized way:

> *"For beginners, [they] look at it and then they'll just think it's like nitpicking and stuff, but I guess it contributes to the overall readability of the project. So anyone who reads it even, he can just read it in one go, he won't have to like switch between styles and stuff" (P2).*

On the other hand, this emerges as a source of rejection for new contributions and conflict between project leaders and new contributors who may not be familiar with the project's style. P1 recounts that as a new member of the Drupal community, they "would be kinda worried" about contributing to the project, "because maybe even if your code works, they'd shoot you down for not having tests or not following some kind of rule or something... Maybe even unwritten ones" (P1). P1 suggests that this raises the bar very high for first time participants, who may not be comfortable to contribute when faced with such tough rules.

Conflicts can also occur between more experienced contributors who have different beliefs about how code should be written. A common source of debate concerns writing tests. P9 recalls that when they first publicly announced an open source project, someone wrote on the Google Group: "When you push out open source software, you need to write tests" (P9). For P9, on the other hand, the project was a hobby:

> *"It is something that I am passionate about. I wanted to share it, so I didn't write any tests. The guy was a bit adamant that I should*

write tests, otherwise I should not ever push out any code into open source. So we had a bit of a disagreement on that." (P9)

In this case, because the contribution was modular, and there was no maintainer to seek approval from, it was easier to disagree:

"The end resolution was I suppose that he kept to his opinion, and I kept to mine, so, I wouldn't even call it we agreed to disagree. It's just that after a while we did not respond to the thread anymore." (P9)

Though it is rare to find process conflict concerning responsibility in FOSS due to its' voluntary nature (P8), process conflicts may also emerge when credit is not adequately given to maintainers, or if a contributor does not respect the team's hierarchy of contributions. For instance, when Debian developers with commit access wish to modify code they are not themselves responsible for, they are expected to follow an unwritten rule (P3). Specifically, committers need to submit their Non-Maintainer Upload, or NMU, with a delay of 7 days to allow the maintainer of this code to review and accept the changes. If code is uploaded without adhering to this convention, Debian maintainers can get very upset (P3).

Because every FOSS project has its own management style, P1 suggests it is up to new contributors to "figure out the correct etiquette for contributing code", for instance, finding out whether "maintainers would only accept patches that had test coverage" (P1) and avoid conflict. In absence of formal style guides, participants suggest leaving a comment on the issue asking about the correct procedure and attribution to demonstrate due diligence. When conflicts regarding authority do occur, they are likely to transform into affective and/or normative debates. The next sections will cover this interplay and its impact in greater detail.

Taken together, process conflict appears to involve issues of logistics rather than responsibility in FOSS teams, such as the style of writing code. The disagreements may stem from lack of familiarity, or conversely, alternative systems

that are too entrenched, but appear to work towards a common direction for the team going forward.

3.3.3 Affective and Transforming Conflict

The previous sections have already suggested that both task and process conflict are likely to transform into affective conflict over time. In fact, affective conflict was not mentioned in isolation by any of the study participants – rather it accompanied other issues already discussed. To recap, affective conflict involves interpersonal disagreements such as personal attacks between team members. Findings show a number of different kinds of conflict transformations, thus this section will present examples of such transformations, and discuss their significance with relation to the changing nature of ongoing teams.

Firstly, findings highlight the transformation of task into affective conflicts (P5, P2, P4). Often, the affective conflicts described were the result of already mounting frustrations with a task, or involved a power imbalance between disagreeing parties. For example, the lead of the Zope project approached one of his maintainers after a conference and chastised them personally instead of the team for not being more careful when making changes to the code base (P5). P5 suggested that this task conflict turned personal because of already mounting frustrations during the conference about the lack of progress in the project in general:

> *"And in part it was a reaction to my negative sort of attitude, [and] all this frustration behind me already to get this small progress. [...] So we had this sort of discussion and we had been friends [for] years by then [and] that was the last time actually I met him in real life". (P5)*

Participants have also reported examples of procedural issues that evolve into affective conflict. For example, following a prominent incident in which the

Linux kernel was hacked, a disagreement emerged between two leaders, Linus Torvalds, and Shawn Pierce, about implementing tighter contribution security in their software version control system (P2). Though the task at hand to improve security was clear, the leaders disagreed on the procedure to accomplish this. The conflict escalated to a relational one, when Shawn suggested an alternative solution to the one Linus proposed, and "Linus said 'No, totally wrong'. Except he didn't use those words, he used something worse". (P2)

According to P2, Linus Torvalds' harsh tone had transformed a procedural issue into a relational one, similar to the task-relational conflicts above. Interestingly, participants often reported relational conflicts emerging from task or process conflicts with authority figures (P5, P2). P2 suggests this may be because project maintainers and leads have greater demands on their time, and are also in a position of authority that allows them to "get away with" being more offensive.

Taken together these findings indicate that in FOSS teams, task and process conflicts can evolve a relational dimension over time due to the ongoing nature of the teams, as frustrations about unresolved task issues build momentum:

> *"Mostly . . . just technical debates and what, but sometimes it can get quite degrading and down to like random name calling. . . There are some people who- sometimes there are these sarcastic remarks that are passed. And then there are some people who are very very sensitive to those things, and then it just starts exploding" (P3)*

As the previous section on process conflict has shown, some FOSS projects have a 'way' of doing things within the group that may be unwritten, such as a style of writing code or accepted practices for division of labor. Disagreements about a task can also evolve into procedural conflicts when developers are not aware of this style, or have a strong opinion about doing things differently.

For instance, a maintainer for the Zope project sparked a conflict when they removed a list of dependencies without discussing the decision with their team-

mates (P5). The other maintainers disagreed with the decision to remove the dependency list, in essence a task conflict. However, when it later emerged that proper procedure was not followed, and this change was not discussed with other members who may still depend on this list, the conflict became procedural. P5 attempted to revert the change, in essence fixing the task element of the conflict, but interestingly they faced criticism from the community for rolling back the changes without discussion as well.

In general, the above section has shown that ongoing virtual teams such as FOSS development teams exhibit the tendency for conflict types to transform and evolve over time. In addition, more combinations of conflict transformations have been observed than those documented in previous work (Arazy et al., 2013). In particular, conflicts involving project leaders or an imbalance of power appear to frequently take on a relational dimension. Thus while ongoing virtual teams may evolve hierarchies and procedures to help them manage their community growth (O'Mahony & Ferraro, 2007), this may introduce new tensions in the project over time.

3.3.4 Normative conflict

The previous sections dealt with conflict that emerged due to misunderstandings about the coding task to be performed, project styles and procedures or interpersonal friction. The following section deals with conflicts that are qualitatively different from any of the above three classifications. They are referred to here as normative conflicts, as they involve disagreements concerning group norms and values that may be entirely disconnected from the day-to-day action of working together. They may emerge either out of a perceived dissonance between the actions of a team and its stated values or standards for behavior, or when such standards and values are undefined or ambiguous. Normative conflict may therefore involve disagreements about issues like changing project policies, ethics, and overall project ideology such as attitudes towards minorities

and license debates.

Prescriptive and descriptive norms

A number of reported normative conflicts emerge out of a perceived dissonance between the stated aims or values of the project (its prescriptive norms) and actual practice (its descriptive norms) that are explicitly highlighted by one or more members of the group. For instance, following an incident where a few users were banned for being abusive on the project mailing list, Debian developers debated whether the practice of banning users was going against the community norms of free speech:

> *"There was a bit of a debate about [the] trigger happy banning behavior of some people. Some people were concerned that freedom of speech was being obstructed" (P3)*

The Debian community was not only debating its own policies with respect to banning of users, developers were also concerned that in doing so, the community was no longer living up to their own beliefs concerning free speech. The descriptive norms of banning individuals were thus misaligned with prescriptive norms of free speech in the community, leading to a manifestation of normative conflict. The Debian community recently adopted a Code of Conduct in response to normative conflicts like this one, clarifying acceptable behavior and actions that will result in being banned.

Similarly, in a high profile incident within the Ruby on Rails project, a community member resorted to hacking the project to demonstrate a dissonance between community norms and practices.

The Rails community has strong opinions on how the programming framework should work and be used (P1, P4): "Rails is opinionated software. It makes the assumption that there is the 'best' way to do things, and it's designed to encourage that way." (*Rails Guides*, 2014). The "Rails way" is the group's

ideological construct that helps to align the efforts of contributors towards the same goal. It emphasizes "convention over configuration".

A developer (Homakov) uncovered a potential security flaw that emerges when a developers do not follow one of these Rails conventions. Homakov presented it to the community, however:

> *"He was pretty much ignored and put down because the core team thought it was a non-issue and that things just weren't done that way because it's not the Rails way" (P1)*

To illustrate the severity of the security flaw and that this convention was not being followed even in large projects, Homakov chose to exploit the vulnerability to make unauthorized commits to the Rails project on GitHub. He also spoofed comments by prominent members of the Rails team, such as DHH, the project lead. Through this hack, Homakov pointed out that the prescriptive norms of Rails as the "best way" to write software were at odds with their actual practices because they overlooked the severity of a security flaw in the framework.

> *"Perhaps if he didn't feel that his point was worth proving we would be dealing with an insecure by default framework just because the core maintainers or a majority of them don't think that's 'the way' to do so". (P1)*

The event put a great deal of time pressure on the Rails security team, who worked to release a more secure version of the software (P8). Normative conflicts therefore have powerful implications for the team as a whole because when they erupt they force the group to evaluate their collective values, and if necessary, make significant changes to ensure group practices continue to be in line with their aims.

The above event highlights a few interrelated conflict types. The conflict erupted because Homakov showed that the norm of convention over configura-

tion, while deeply embedded in the Rails community, was also a potential security risk. Thus the issue began with a normative conflict. Through Homakov's frustration with lack of response to his finding, the conflict gained an affective dimension. By resorting to hacking to demonstrate the severity of the issue, the conflict also acquired a process dimension because Homakov violated the agreed upon procedures for introducing changes into the project and created time pressure for the core team to address the issue.

Clarifying ambiguous norms

Normative conflict may also arise out of a need to clarify an ambiguous standard of behavior within the community. For example, a Mozilla developer published a controversial article about marriage that was syndicated on Planet Mozilla. The post offended some members of the community as well as the public at large, and a debate erupted concerning whether the platform was an appropriate place for publishing non-Mozilla related views (P7).

One participant wrote: "Can you please keep your personal prejudices off Planet Mozilla?", while another defended the post: "Absolutely not. Planet is for all kinds of discussion from the Mozilla community. While I don't agree with [the author's] stance here, I fully support his right as a community member to post his thoughts on Planet. Feel free to ignore posts you don't like."

A broader debate emerged as a result of this incident about whether and how personal opinions should be accepted on the Planet Mozilla platform. Some members highlighted precedent in which earlier attempts at censorship were severely criticized by the community. The relevant norm was eventually clarified by one of the core Mozilla team members, explaining that Planet was created with the intention to encourage all content, including personal, and that this was documented, albeit obscurely, in its foundation documents.

The Opal community presents another example of clarifying community values of free speech through normative conflict. A GitHub issue was created

within the Opal project proposing the removal of a maintainer who published "transphobic" materials on their personal blog. The issue was hotly debated by the community. Due to the public nature of GitHub discussions for free accounts, the debate also included a large number of outsiders.

The vision of wanting to create an inclusive community and punish the offender appeared at odds with the project's belief in free speech. It also conflicted with a more practical need to focus on code contributions rather than personal opinions in order to continue to encourage open participation in this voluntary ecosystem. This created an ethical dilemma of whether and where to draw the line on censorship in favor of more diversity or more contributions. That is, it involved a reformulation and clarification of community norms.

In response to the conflict, Opal developers drafted a clear code of conduct outlining a standard for behavior in the project. The community emphasized this new behavioral standard will continue to be iteratively improved through feedback.

The Opal normative conflict also fuelled the development of a separate Contributor Covenant (Ehmke, 2014), an online movement that aims to standardize codes of conduct across different FOSS projects, and emphasizes anti-harassment policies. Interestingly, the Opal project opted not to adopt the Contributor Covenant, and developed their own community norms instead.

Coordination activities, such as deciding on a code of conduct, tend to be deferred in FOSS projects because they require more interactivity than usual development practices which afford more modularity and sequential interdependence (Howison, 2009). For example, the Ruby community has historically followed an informal and unwritten code of conduct they referred to as "MINSWAN", that is "Matz is nice so we are nice". Matz is Yukihiro Matsumoto, the creator of the Ruby language and founder of the project, who is known for his genial approach and warm attitude. The community thus organically modelled their behavior toward each other based on the example set by Matz. Partici-

pants from the Ruby community reported that conflicts rarely arose in the Ruby project because, upon seeing any potential disagreements, community members reminded one another of the "MINSWAN" motto and tried to resolve discussions amicably.

However, FOSS projects are very open systems and thus even the relatively peaceful Ruby community eventually found itself involved in a normative conflict initiated by an outsider. In January 2016, the creator of the Contributor Covenant approached the Ruby community about ratifying their code of conduct. By this time, the Contributor Covenant had been adopted by a number of large and visible projects, such as Rails, Angular JS, Mono and Apple Swift. The Covenant creator submitted an issue request to the Ruby core team, assigning it directly to Matz, to consider adopting the Covenant for the Ruby project as well as it "came from and has been so widely adopted by the Ruby community at large" (Ehmke, 2016).

While some members of the Ruby core team responded favorably in support of the request, others had strong objections. Some cited a lack of need for a formal code of conduct, as an informal one already exists. Others had more specific objections about the Contributor Covenant itself, and that it fails to reflect the nuances of the Ruby community that make the project attractive to developers. For instance, one developer suggested:

> *"Personally I feel that Aaron, DHH or in fact any other prominent and respected member of the community is more than capable of writing a Ruby Mission Statement that better reflects Matz's intentions." (J, 2016)*

Another developer opined, "This is how projects die", out of concern that introducing a code of conduct would disrupt an ecosystem that has been working well so far (S, 2016).

In response to this disagreement, Matz put forward for discussion a short Code of Conduct that he felt was a more appropriate reflection of the community

norms. Though Ruby is a benevolent dictatorship[4], Matz opted to discuss the proposed formalization of norms in an open thread, to allow the community to give feedback and ensure the proposal is consistent with feelings within the community.

In the above examples, normative conflict serves as a mechanism that encourages teams to discuss and formalize more explicit norms at a moment in team development when it becomes necessary to do so. In the case of Opal, an actual instance of hate speech prompted the discussion of how to develop group norms to prevent this while retaining the flavor of the community. In the case of Ruby, the community did not experience a critical need for a formalized set of behavioral norms until these were suggested from the outside. However, because the suggestion resulted in a significant disagreement, it became necessary to address this concern and also introduce a code of conduct that was reflective of community sentiment.

At the same time, as the previous section on prescriptive and descriptive norms has highlighted, project norms can become out of date as communities grow organically. Normative conflict therefore also acts as a mechanism that highlights the disconnect between existing community practices and norms and those intended, allowing projects to face the issue. However, as next example shows, it may also lead to turnover because the conflict may highlight fundamental differences in the way individuals think about an issue.

Node.JS - An extended normative conflict example

To conclude this section of the chapter, one extended conflict example will be presented that illustrates a number of the above themes. Specifically, the episode shows that normative conflict can interact with other conflict types already presented, and result in significant structural changes for the group. The following conflict episode was independently discussed by a number of participants (P5,

[4]A "benevolent dictatorship" refers to a FOSS project with a strong leader figure who can make unilateral decisions without consulting other community members.

P4, P10, P13), and simultaneously involves procedural, affective and normative dimensions.

In late 2013, a developer submitted a pull request to libuv, a sub-project of Node.JS, replacing a gendered pronoun inside a line of documentation with a gender-neutral alternative (P5). Ben Noordhuis, the project maintainer, rejected the pull request with a short comment that the patch was "too trivial to bother" (P4). While for Ben the source of the disagreement was procedural, an issue of time management, the community saw this as sexist, and "went a bit crazy at him about it" with personal attacks and flames (P4). Thus the conflict evolved an effective component.

While the patch may appear to make a valid point, it goes against community standards for pull requests which ensure efficient coordination and use of developer time:

> *"From the perspective of outsider, that looks like a logical thing, but from the perspective of a programmer, (there is more) than just a logical reason, there is a standard for issuing a pull request [. . .] Maintainers do not accept corrections for minor formatting – in order to accept an issue it needs to be well documented and also (have) some technical reasons" (P13)*

Many projects reject pull requests that only fix grammar issues, unless the submitted change makes substantial contributions towards improving the documentation: "I know that in other projects (as well), people do not usually accept this kind of request." (P9) Thus commits that do not justify their inclusion on technical merit usually get rejected, particularly in larger FOSS projects where maintainers have greater demands on their time (P13).

Vocal public criticism of Ben's decision highlighted a normative conflict in which the community standards for accepting code were not in line with behavioral standards for gender equality. To address this normative issue, Isaac Schlueter, a former community leader, stepped in and accepted the pull request

without any further discussion, thus undermining Ben's authority as maintainer (P5). Ben, in turn, reverted Isaac's commit:

> *"He [Ben] didn't revert it because he disagreed with it, he reverted it because [they] have procedures for getting stuff into libuv. And although Isaac has commit rights, he shouldn't [...] get around the process." (P4)*

For Ben, the issue continued to be about procedure for getting code into the library, and accepting a pull request in direct violation of the library maintainer violates procedure. Isaac, on the other hand, viewed the issue as a normative one, arguing that he was upholding the norms of the project, to be "inclusive of non-male people and that our language should reflect this explicit inclusion" (Schlueter, 2014). Thus different parties participating in the conflict may have had different perceptions of the source of the conflict they were involved in, resulting in the "mixing up of a social issue with a technical issue" (P13), or a series of "different truths" for participants.

Ben eventually resigned from the project due to community and public pressure. In particular, a public blog post from a company that owns the Node.JS trademark chastised Ben's response and suggested that if he were a Joyent employee he would have been fired.

Furthermore, the interplay of different third parties to this conflict (the Joyent organization and a former project lead) stimulated the community to reflect on its ties and dependency on for-profit organizations in the community. Faultlines formed between supporters of an independent foundation and those who believed in the value of organizational contribution. Eventually, part of the the project forked into IO.JS, to continue the work without the influence of external organizations. Interestingly, Ben returned to the new IO.JS community and worked alongside Isaac to develop the fork because they were united in an anti-corporate agenda. Node.JS and IO.JS development efforts proceeded in parallel for several months, until Joyent proposed to form an independent

Node foundation to solve the governance and corporate influence issues that had initially led to the fork. After some negotiation, the IO.JS team decided to rejoin the Node.JS project under the new terms of developing a foundation and to reduce the amount of duplicate work that was being put into both projects.

As the above example demonstrates, it is possible for localized conflicts, such as about group processes, to bring to light higher order issues in the project that cause faultlines in the group. In the case of Node.JS, this conflict allowed the community to clarify several community norms, such as inclusivity and when it is appropriate to revert others' commits. At the same time, the conflict raised broader issues about governance and independence, led to a fork and the eventual establishment of a foundation to oversee project development.

Taken together, the above normative conflict examples highlight its power in driving forward team development in ongoing virtual teams that recursively evaluate and adjust their techno-social structures. While causing friction in the short term, normative conflict may have potential to enact significant transformation in the long-term through clarifying norms, values and vision.

3.4 Discussion 1

The above findings address two of the research questions posed at the beginning of this chapter. Specifically, RQ1 was concerned with understanding conflict experience from the perspective of FOSS participants, and the extent to which existing conflict theory is applicable to such self-organizing virtual ongoing teams. RQ2 was interested in identifying the mechanisms through which conflict influences future team structures and inputs.

Addressing RQ1, the findings show that FOSS teams experience four different kinds of conflict – conflict about task issues such as feature road-maps, implementation and dependencies; process conflict about coding style and attribution; affective conflict involving interpersonal incompatibilities and emo-

tional exchanges; and normative conflict arising out of conflicting or unclear group values and standards of behavior. These map to the intragroup conflict taxonomy proposed by Jehn (1995, 1997) well, with the exception of normative conflict that emerges as an additional factor for consideration. Furthermore, in support of Behfar and colleagues' (2011) work, the above sections show a clear distinction in the way task and procedural issues emerge in FOSS communities.

Though in principle task, process and affective conflict in FOSS teams are similar to conflict in traditional organizations, their substantive manifestations differ and reflect the specific context of FOSS development. For example, task conflict for FOSS teams involves debates about proposed features, dependencies or efficient implementations. In other words, task conflict emerges out of disagreements about what to work on. Teams that are less empowered and have tasks assigned to them, such as short-term student teams and hierarchical organizational teams, are more likely to engage in task conflict that clarifies task requirements, or how approach the given task. Similarly, while process conflict in traditional organizational teams is concerned with elements such as task assignment (contribution), the voluntary nature of FOSS teams makes it difficult to hold expectations about any time invested. Thus process conflict is more often concerned with logistics of performing tasks, such as coding style and giving credit. These findings suggest it is important to keep in mind the context of teams being examined, and to ensure validity, develop measurement tools that are meaningful for the participants' context. Furthermore, the findings suggest the utility of viewing process conflict as multi-dimensional (Behfar et al., 2011), that is, consisting of both contribution and logistics dimensions, as different types of teams may place emphasis on different elements of procedure.

Interestingly, affective conflict in this study is not reported in isolation. Rather it is a consequence of escalating existing task, process and normative issues. This is consistent with work that suggests the likelihood of conflict transformations (Arazy et al., 2013; Jehn, 1997; Martinez-Moreno, González-

Navarro, Zornoza, & Ripoll, 2009; Simons & Peterson, 2000). Furthermore, some of the examples presented above include two or more kinds of conflict that evolve over time. Taken together with earlier work, these findings suggest it may be relevant for conflict management to understand and focus on managing the trigger point at which conflicts evolve into larger, more protracted and/or unproductive discussions.

Second, the findings highlight that ongoing virtual teams experience an additional type of conflict that Jehn's taxonomy does not consider – that of normative conflict, or conflict concerning the values and expected behavior of the group. Normative conflict manifests in two distinct ways. First, normative conflict emerges when the descriptive norms of a group (what the team actually practices) fall out of line with the prescriptive norms (expected standards of behavior), a pattern consistent with predictions made by Packer (2007) for long-term social organizations. Second, normative conflict emerges when standards of behavior are not well defined, or do not yet exist all. This type of conflict may be the direct result of the on-going and self-determining nature of FOSS teams. Because ongoing teams have more time to evolve than temporary virtual teams, they are more likely to develop distinct group norms (Saunders & Ahuja, 2006). In fact, K. J. Stewart and Gosain (2006) found that FOSS teams develop their own community norms and ideologies, and that these can vary significantly from project to project. Additionally, FOSS teams are a recursive public (Kelty, 2008): they are able to not only view and modify the source code to their software projects, but also reflect on and adapt their own practices and means of production.

Findings on normative conflict therefore address RQ2, and illustrate how normative conflict impacts future team structures. Manifested normative conflict brings to the surface a lack of clarity in behavioral standards or inconsistencies between group values and actual practices. Thus, a resolution of these issues requires the clarification or establishment of new social structures that

provide a basis for modelling group behavior. Examples of resolutions include community leaders providing definitive clarifications in times of uncertainty, as well as the establishment formal codes of conduct offering consistent behavioral standards. Thus normative conflict can result in changes to the social structures of the group, such as group norms and culture.

Normative conflict was also found to crystalize factions of opinions, such as in the Node.JS example, and thereby create faultlines within the community of developers with opposing issues. While all forms of conflicts may potentially encourage contagion, that is, for developers to take sides and form coalitions (Coser, 1957; Jehn et al., 2013), normative conflict represents issues that concern the entire group. Thus it is more likely to lead to coalition formation as more members weigh in on discussions. Discussions in virtual team settings are also more likely to be available to the whole team and archived, and thus have a greater likelihood of reaching more team members.

Taken together, normative conflict emerges as a unique and important additional factor in the intragroup conflict taxonomy. It is particularly relevant in ongoing virtual team settings, and has the potential to have far reaching effects not only on immediate team performance but also future team structures.

3.5 Conflict Antecedents

The final research question for this chapter (RQ3) was focused on understanding factors that may contribute toward conflict manifestation, or conversely, help to reduce conflict levels. The following section briefly describes four themes that have emerged during coding that were associated with either increasing conflict occurrence or helping its management, based on participant reports. They are geographical distribution, leadership style, distribution of decision-making, and team interdependence. The chapter concludes by discussing their implications for the study of conflict in FOSS, and virtual teams more broadly.

3.5.1 Geographical Team Distribution

Though FOSS developers occasionally meet face-to-face to encourage community building, most important decisions are made on archived mailing lists or internet relay chat (IRC) meetings to allow the whole distributed team to participate. In practice, this presents some challenges.

Several participants have reported conflicts arising from communication issues such as different time zones, having too many communication sources to monitor, information overload, and difficulties being heard (P1, P4, P3, P12, P7, P13). Participants noted that they have to stay up late at night in order to interact with team members in Europe and North America (P12, P7, P3). For many projects, IRC, a synchronous communication tool, is used to co-ordinate important issues such as obtaining feedback on a proposed feature before making a formal pull request (P4) or to ask more experienced contributors questions (P12, P7). Some participants express frustration that developers from Asia and Australia need to conform to North American time zones (P12, P7, P3).

Developers also report conflicts arising from communication overload and contributions "getting lost in the noise" (P4). For instance, particularly in larger projects, a steady flow of communication comes through the mailing lists every day as subsequent time zones come online:

> *"[You] can't subscribe to everything these days, [and] can't read everything subscribed. [By] the time you're done a whole bunch of other e-mails come in. [It's] a whole global thing [...] people go to sleep, the next time zone wakes up and starts replying." (P3)*

Additionally, projects may use multiple communication channels for decisions that are not always well publicized, resulting in an impression that decisions happen "behind closed doors" (P1). For example, the inability to follow all possible communication channels has left the packager out of an important decision regarding the application they are maintaining:

"There was a decision that happened at some random IRC session that I didn't know about. [...] One day I just woke up [and] I read on some blog somewhere, "[My project] is no longer default". I'm like, "Huh?"" (P3)

P3 was unable to weigh in on the decision to select a default media player for the operating system, and another package was chosen instead. P3 raised the issue on IRC, linking notes from the missed meeting. An affective conflict erupted when other supporters expressed their solidarity by "spamming [flaming] the whole meeting notes upside down" (P3).

Thus, taken together, greater geographical distribution among participants, coupled with a heavy reliance on computer-mediated communication leads to greater levels of procedural and affective conflicts in FOSS teams.

3.5.2 Leadership

Although FOSS projects are often presented as an example of self-organizing systems with relatively flat hierarchies, the present study's findings show that a sense of leadership represented by one or a group of experienced developers can work towards reducing conflict occurrence. A stronger sense of authority may prevent conflict by encouraging alignment of opinions with a more experienced developer. Leaders may also serve an interpretive function during conflict episodes to bridge gaps in understanding between contesting parties. Finally, leader personalities emerge as a source of community norms and best practices.

A common source of conflict identified earlier involves proposing an idea or implementation, and receiving criticism from community leaders or maintainers concerning the idea (task conflict). Participants suggest in such cases it pays to defer to the experience of those in charge of accepting pull requests, as they are likely to be drawing on greater experience with the project and code base (P2, P12, P7):

"[Conflict occurs] because you think that they are wrong, but based

on your own knowledge which isn't as much as them. So they might have taken into account other factors that you haven't. So usually they are right. You think that they disagree with you, but it's... not personal" (P2)

Accepting such criticism involves a sense of trust in the leader's experience and perspective:

"There are some things that are quite clear cut, then yeah he [maintainer] just makes the decisions. And most of the time they are right, because yeah- he's a smart person, so we trust him." (P2)

Therefore, particularly in ongoing virtual teams that rely primarily on text-based communication (such as e-mail or IRC) for coordination activities, it is useful to provide social cues that indicate developer of a developer: "You have a sense that OK somebody big is reviewing my code. It's got to mean something. So that's sort of a different situation now" (P7). Furthermore, platforms such as GitHub are useful because they make levels of experience more transparent via developer profiles, emphasizing leader roles.

In situations of uncertainty between developers, a leader may also serve an interpretive function to clarify the "different truths" each side is adhering to. P6 explains the strategy of "quiet talk" by an authority figure with some level of gravitas serves to resolve an issue by highlighting to the participants involved "here's what you're not seeing" about the other's perspective (P6). For instance, P7 recalls getting defensive when they first proposed an implementation to the Mozilla community and received critical feedback from the IRC channel, resulting in conflict. When the criticism got "very loud", the lead developer initiated a private chat with P7 to explain "what people are actually saying"(P7). The leader was not only able to interpret the criticisms in a way that allowed the newcomer to understand, but also served to encourage and support the contributor to persevere in significantly improving the implementation (P7).

A project leader or founder's personality may also affect how conflict and disagreement are dealt with in the community (P9). For instance, the Ruby community's motto is "we are nice because Matz is nice" (P10). Thus, whenever the community approaches conflict, someone invokes this phrase to indicate that the community should not argue about this, resulting in a less "abrasive" community (P9, P10). Ruby on Rails, on the other hand, is a framework based on Ruby but with very different community norms. Rails is "opinionated software" which means it suggests certain conventions for writing software, because its' creator, DHH is also "very opinionated" (P9). Thus in situations of disagreement, the community can often drawn on convention or for DHH to weigh in with opinion for a resolution. P10 explains that a leader with strong opinions is good for a FOSS community because this provides a concrete direction for project growth and success:

> *"In a lot of cases, what are you banking on? You are banking on the authors opinion, because he does things in a certain way, and you find there is value in doing it in certain way. This is why everyone uses Rails. When DHH comes out and says [something], it is the Holy Bible. It is his project, it is his opinion that counts." (P10)*

Thus even though FOSS development is often represented as an egalitarian form of 'peer production' in which all are equal, a strong sense of hierarchy and respect for authority emerges as a recurring theme in preventing disagreements among FOSS developers.

3.5.3 Distribution of Decision-Making and Bikeshedding

While strong leadership emerges as a powerful positive overall force on FOSS teams in preventing conflict, more dispersed decision-making was conversely identified by participants to lead to more conflict and less ability to reach consensus. In particular, the transparent and public nature of FOSS team discus-

sions allows anyone outside of the project to weigh in on discussions, and this can result in smaller project issues growing into large public debates.

Projects like Debian allow one equal vote per official contributor. However, this distributed decision-making system makes it very hard to come to consensus (P3). For example, a debate about whether to replace the boot sequence of the operating system (the Init daemon, or init.d) with two viable alternatives (Upstart and SystemD) met with stalemate lasting several years as "Upstart proponents like to mud-sling about SystemD and the other way. And neither really tries to understand the other side of the picture" (P3).

The debate resurfaced within the project several times over the course of several years, and the project came to a resolution only after the issue was promoted to a more authoritative technical committee to make a final decision (P3).

The transparent communication nature of FOSS discussions further problematizes this dynamic because it can invite public participation. Debates become lengthy as a greater number of people weigh in from the public, while the relative empowerment of community members in distributed decision-making systems encourages "commenting for the sake of commenting" (P7,P10). This phenomenon is known within the community as "bikeshedding". bikeshedding is the tendency toward expressing an opinion on trivial, but accessible, issues, such as the right color to paint the bikeshed. More complex decisions, on the other hand are often met with few opinions from the general public (Kamp, 2003). As a result, simple issues that are open for public debate receive more attention and fail to reach consensus, while more complex issues do not benefit from enough discussion because they are harder for most team members to understand.

For example, P7 recalls a conflict episode that emerged when a Mozilla developer unintentionally borrowed code from an external source that included a tracker. Mozilla publicly clarified that the practice was not acceptable, and "tracking is very bad" (P7). However, the debate that followed the incident

included not only the whole Mozilla community, but also the general public through blogs and comments: "the online criticism, like within the room, became a sort of a Internet kind of criticism" (P7). Similarly, a British Ruby Conference was called off several years ago because of strong public criticism on Twitter of the organizers, who invited an all white male panel of speakers (P10). Examples presented earlier such as the Node.JS gender commit, and the backlash against Opal also suffered from bikeshedding because their projects were publicly hosted on Github and thus all discussions occurred in a public forum.

To simplify participatory decision-making, some communities have adopted a more streamlined voting system for issues (P11). For instance, the GeoTools project and some projects under the Apache Foundation utilize a +1/0/-1 vote system. To prevent bikeshedding, a -1 vote requires a proposition of an alternative idea or implementation. Thus while overall distributed decision making may generate more conflict, this can potentially be managed with more formal rules for how to give feedback, such as the requirement to suggest an alternative when down-voting an idea.

3.5.4 Interdependence

Finally, findings show that the amount of conflict experienced in a team varies with the level of interdependence in the project.

P4 points out that the Node.JS project is very modular, and therefore, "if you disagree with someone's choice, you can [just] replace the piece". "In Node, there is less need for people to all agree on something because if you disagree with someone's implementation you can create your own replacement." (P4) On the other hand, the Node Package Manager that ties together all these distributed modules (NPM), is more centralized in it's code architecture design, and:

> *"[It] has all of the same problems as [you would] expect [...] a whole bunch of people who have different concerns, different prior-*

ities and different needs out of a piece of software [. . .] They're all like vying to get NPM to suit their needs. Now if NPM followed the same distributed architecture. . . maybe it was made up of a bunch of different modules and you could kind of roll your own NPM that would possibly be a better solution" (P4)

Thus, because a project's technical structure is more interdependent, their social structure also experiences greater friction.

P6 explains that the relative modularity of the code base is sometimes determined by the project function, and cites a long-standing conflict between Linus Torvalds, the creator of Linux, and his mentor, Andrew Tannenbaum. The Linux kernel supplies device drivers and in essence connects the software layers of FOSS operating systems to the computer hardware. Although it consists of many individual pieces, they are all interrelated (P6). This is because there is a strong emphasis on performance in the project on account of Linux being a fundamental piece of the operating system. Thus Linux runs into constant issues where a contribution by one developer is able to break the entire function of the kernel (P6). This is a recurring point of criticism by Andrew Tannenbaum, who frequently engages in conflict with Linus and other project maintainers about the architecture of the kernel. However, Linus argues that the kernel cannot be designed more modularly when performance is the focus, thus a certain amount of interdependence and therefore technical conflict is necessary (P6).

3.6 Discussion 2

With the final research question, RQ3, the present study aimed to identify conflict antecedents that may stimulate or help in reducing conflict levels in FOSS teams. The above findings highlight four distinct factors, namely the extent of geographical distribution in the team, leadership, the extent to which decision-making is distributed, and the level of interdependence in the team. The impli-

cations of this for research and community management are discussed for each factor below.

First, the present work highlights that similar to more traditional virtual teams, conflicts in FOSS teams can arise directly out of their technical constraints, such as communicating across time zones via multiple communication channels (Mortensen & Hinds, 2001) and the level of the team's interdependence in their work (Kankanhalli et al., 2007). However, findings also show that different kinds of interdependence may be relevant for projects serving different functions – projects that place an emphasis on performance tend toward more interdependence, and as a result may need to learn manage greater manifestations of conflict.

Second, the findings highlight the important role of leaders, even in voluntary distributed settings that are known for their self-organizing nature, such as FOSS projects. Leaders are either project founders, individuals appointed by an outgoing leader, or core developers whose work is central to the community. Madey, Madey, Freeh, and Freeh (2002) refer to this group as lynchpin developers because they are often responsible to a big chunk of work and decisions made in projects. Specifically, leaders in FOSS teams play a big role in the overall culture that evolves in the project – the extent to which a team is conflict-averse is connected with the personality of project founders. More generally, leaders exert an idealized influence on the team (Bass, 1990), that is, they lead by example, and maintain their status at the forefront of the community by committing rich and complex code. In other words, FOSS leaders provide a role model for other developers to follow by "walking the talk". This aligns closely with the transformational style of leadership that is often studied in traditional organizational settings in connection with positive team outcomes.

FOSS leaders also appear to serve a useful function in conflict resolution by using their idealized influence to step in and make quick decisions to resolve conflict. They serve as arbitrators, bridging the gap in "different truths", or

understanding among developers, that may lead to conflict. Community leaders by their very position in the team engender trust, as they are expected to be more experienced in project issues and the code base. Thus it may be useful to display team member experience with tasks on their profile, as this added social cue can serve to help team members judge the credibility of feedback and criticism given.

Taken together with the above findings on the similarities between conflict types experienced, the present work suggests it is useful to draw on insights from work on traditional organizational teams in understanding conflict in ongoing, voluntary virtual teams such as FOSS teams. There are also some important differences that emerge due to the unique voluntary and virtual context of FOSS projects.

For instance, FOSS teams vary in the amount of decision-making power contributors have toward the overall project direction. Some projects are "benevolent dictatorships" where the project leader/founder has a final say in decisions, while others are run by a steering committee, or practice a one-vote-per-developer policy. As a result, conflict emergence and resolution also differ. On one hand, a distributed decision-making structure supports the high virtuality of FOSS teams and alleviates the bottle neck of awaiting a deciding answer from one or a small number of individuals on important matters. Instead teams with distributed decision-making styles operate on rough consensus among the individuals closest to the problem at hand. On the other hand, participant accounts highlight that allowing more individuals to participate in decisions often leads to protracted conflicts because as the number of opinions increase, so does the tendency to identify and argue against the consequences of some decisions. In particular, distributed decision making systems have trouble choosing between complex alternatives that have both advantages and disadvantages, or are roughly equivalent, as a vocal minority is often voting against some of the issues.

The transparent nature of communication in FOSS teams also affords greater participation in issues that are not directly relevant to the concerns of all developers. For instance, a disagreement between two parties on a choice of dependency may attract the attention of other developers, either forming potential coalitions, or proposing even more solutions, and thus making the conflict episode more complex. This is particularly true of issues that are more accessible, such as minor changes or discussions about social norms, and thus the community may find it difficult to make progress on these discussions. On the other hand, difficult issues that may require more cognitive diversity do not necessarily receive it. This phenomenon is based on Parkinson's law of triviality (Parkinson, 1957), and known within the FOSS community as "bikeshedding". Bikeshedding has important implications for our understanding of conflict in FOSS teams, and the design of computer-mediated systems.

Prior research on open collaboration has argued that transparency is a crucial element of the implicit coordination that underpins distributed collaborative work (Bolici et al., 2009; Dabbish, Farzan, et al., 2012). However, transparency is a double-edged sword. Towne, Kittur, Kinnaird, and Herbsleb (2013) find that highlighting conflictual discussions leads to reduced perceptions of output quality, unless the conflict is resolved. The present findings take this a step further and show that transparency also affords more participation in conflict episodes by parties not necessarily privy to the original issue, thus disproportionately focusing attention on minor issues, while not enough attention is given to more complex problems. This is a fruitful area for future research to explore – how do we highlight important issues for team discussion that would otherwise get lost in a sea of mailing list updates that revolve around relatively more trivial issues?

3.7 Summary

Taken together, both the manifestation of conflict and its emergence in ongoing virtual teams share similarities with conceptualizations from traditional organizational theory. However, there are important differences that reflect the ongoing virtual nature of FOSS teams.

Because ongoing virtual teams have more time to develop group norms and behavioral standards, over time these can become a source of conflict, particularly when are unclear or expose discontinuity between community values and practices. In this way, normative conflict affords ongoing virtual teams the opportunity to reformulate norms and behavioral standards, and thus emerges as mechanism for team adjustment in the face of growth and change. At the same time, normative conflict spurs change in the social structure of the community when it encourages the formation of coalitions, and may lead to the splintering of a community. Furthermore, ongoing virtual teams that rely on transparent communication archives as activity logs may suffer from more protracted and escalated conflicts when parties not related to the original episode take sides.

Through a qualitative exploration, the present chapter has highlighted a number of critical differences in the way conflict occurs in ongoing voluntary virtual teams, such as FOSS development teams. The present work also aims to generalize the above initial findings and construct a quantitative model of conflict in FOSS that validates the above four conflict types and examines their complex effects across a wide variety of projects.

To do so, the next chapter (Chapter 4) will first review literature relevant to the emergent fourth dimension of conflict – normative conflict, unpacking mechanisms through which it may have potential effects on team structure. Following this, Chapter 4 will draw on work on traditional organizational teams to formulate hypotheses connecting the above four conflict dimensions, their structural inputs, and their effects on team emergent states and outcomes in an ongoing virtual team context. Following this, a methodology for this second study will

be presented together with results and a discussion of their significance (Chapter 5). Finally an overall discussion in Chapter 6 will tie up the findings from both studies into a larger framework with reference to current knowledge in the field, while Chapter 7 will discuss the overall limitations of the two studies together with a path for future research.

Chapter 4

Hypothesis Development - Conflict, Inputs, Emergent States and Outcomes

The previous chapter has highlighted examples of dramatic and public departures from Free and Open Source Software projects, such as the recent gendered commit debate that led to the departure of a Node.JS core contributor. Voices within the community have linked these developer departures with a toxic environment created by high levels of conflictual interaction (Sharp, 2013). In fact, a prominent Linux developer, Sarah Sharp, had recently announced she is stepping down from the community due to its high conflict environment (Sharp, 2015). Given the largely voluntary nature of FOSS participation that affords easy departure, and that relatively little attention has been given to the impact of conflict on FOSS team vitality, this is an important area to address. [1]

Participation in FOSS projects is a largely voluntary activity (Crowston et al., 2012). While a number of FOSS projects have received considerable mainstream success and are able to employ full-time developers to work on the software (such as the Mozilla Foundation, and initiatives like the Google Summer of Code), these examples are few even among successful projects. A recent comprehensive survey showed approximately 60% of FOSS developers are unpaid volunteers (Arjona-Reina et al., 2014). At the same time, FOSS teams have

[1] Material from Chapters 4 and 5 is forthcoming in Filippova and Cho (2016)

porous boundaries that afford relatively easy entry and exit, while the virtual nature of team communication makes it particularly easy to disconnect.

Yet FOSS projects cannot thrive without sustained developer contribution. Markers such as developer activity level (Kevin, Annabi, Howison, Masango, & Crowston, 2004) and number of developers (Krishnamurthy, 2005) are critical factors that contribute to team success (Crowston, Howison, & Annabi, 2006). Given the tenuous nature of FOSS contributor membership and that developer time is primary currency in FOSS, it is important to investigate factors that affect developer retention.

A healthy body of work has already examined various aspects of participation in virtual communities. Prior work has shown that the continued involvement of FOSS developers depends on both their individual motivations and team structural factors, such as leadership, ideology and interpersonal relationships among contributors (Xu et al., 2009). Motivation, in turn, can be intrinsic, such as the desire to help others, as well as extrinsic, involving rewards like reputation and career advancement (Oreg & Nov, 2008). Additionally, developers are more motivated if they identify strongly with the team and subscribe to the project ideology (Zhu et al., 2012).

There are also a growing number of studies on conflict more broadly in peer production settings. Prior work has found that in the context of Wikis, there are high levels of conflict in discussions (Collier & Bear, 2012), conflict increases with community complexity (Kittur et al., 2007) and negatively impacts the quality of articles produced (Arazy et al., 2013). In FOSS projects, qualitative studies have shown that conflict is also an important process (Elliott & Scacchi, 2003) and as Chapter 3 highlights, manifests in four different ways – as task, process, affective and normative conflict. However, research has yet to examine the effect of conflict on outcomes in FOSS teams, especially project vitality (Crowston et al., 2012).

This chapter developers hypotheses in relation to a second study (reported

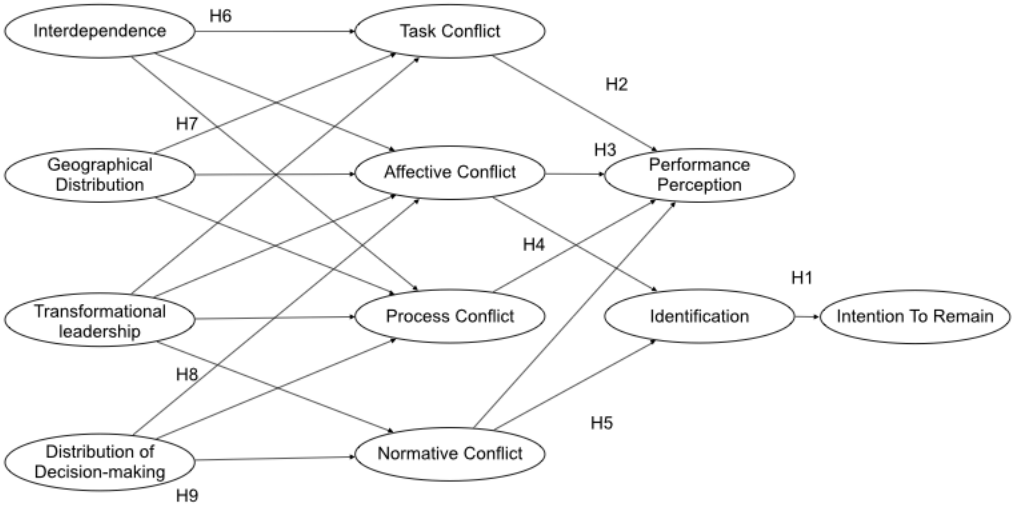
in Chapter 5), which takes the next step in understanding conflict prevalence in FOSS teams by examining the connection between conflict, team inputs and outcomes. Using a large survey of the Free and Open Source Software developer population (n=228), this second study aims to make several key contributions to our understanding of peer production dynamics. First, this study validates the four-factor structure uncovered by Study 1 and, in particular, the relevance of normative conflict in ongoing team settings. Second, the study uses this updated framework to investigate the kinds of conflict that affect developers' attitudes towards their team and their sustained participation. And third, different structural input variables are examined to understand their impact on conflict emergence. Specifically, leadership style, distribution of decision-making, interdependence and geographical distribution are examined because they were highlighted as crucial factors contributing to conflict manifestation in Chapter 3.

In doing so, the present study makes several contributions to knowledge on both open collaborative systems and virtual teamwork. To the best of the author's knowledge, this is the first study on FOSS to quantitatively examine factors that promote conflict alongside conflict's impact on team vitality. The study is also the first to quantitatively examine the emergence and impact of normative conflict in ongoing virtual team settings. Chapter 3 has shown this form of conflict to be a critical feature of community development, and the present chapter goes a step further to chart the role of normative conflict in ongoing virtual team longevity.

Figure 4.1 presents the theoretical model employed in this study. This model follows the classic input-process-output structure and includes both conflict outcomes and antecedents (Crowston et al., 2012). First, the following section will review literature connected with normative conflict and make an argument for its inclusion in an expanded four-factor conflict framework. Then, the chapter expands on the conceptualization of outputs in this work and connects them with the four conflict dimensions. Furthermore, hypotheses are proposed connecting

the four conflict dimensions with possible inputs. Finally, the study also proposes more complex interactions and mediating relationships and explains their significance to the quantitative study of conflict in FOSS teams as examples of ongoing virtual teams.

Figure 4.1: Input-Process-Output Model of Conflict in FOSS Teams



4.1 Hypothesis Development

4.1.1 Normative conflict as a fourth conflict dimension

To address the first research aim of this study, the following section makes an argument for expanding the intragroup conflict taxonomy by including normative conflict.

In prior work, group norms have most commonly been explored as a mediator between conflict and team outcomes, rather than a source of conflict. For example, Jehn (1997) found that group norms about the acceptability of conflict determine whether conflict will negatively impact the team. In other words, groups that expect conflict suffer less from its negative effects. However, group norms are not always homogenous, potentially leading to misinterpretation and conflict among team members. For instance, Jehn (1994) also found that groups

that experience less value consensus (that is, do not share similar values) experience greater conflict.

Prior work suggests that norms can come into conflict due to inconsistencies between the values a group upholds and another existing behavioral standard. Specifically, norm conflict can occur at an internal level (intraindividual) or as an external manifestation of disagreement in the team. Internal normative conflicts are those that occur within the consciousness of an individual, but are not necessarily verbalized, or impact others. External manifestations of normative conflict are those that are verbalized by individuals, and in some way block or interfere with the actions or beliefs of other individuals.

Internal group norm conflicts can emerge because different sets of norms may have different aims that pull an individual in different directions. For example, declarative norms that require bus drivers to both drive carefully and stay on schedule can lead to a contradiction because bus drivers are required to follow both sets of guidelines but may not be practically able to do so (Hamner, 1987). Internal normative conflict may also occur when individuals identify with more than one social group at a time, thus different social realities and expectations may come into conflict (Roccas & Brewer, 2002). For example, while journalistic ethics dictate publishing unmodified photos, artistic and organizational pressures on journalists for more sellable images result in pressure to digitally enhance photos (Lowrey, 2003). In peer production settings, contributors are commonly part of several different projects at the same time (Arjona-Reina et al., 2014; R. A. Ghosh et al., 2002). Thus they may also experience competition between various project norms. Internal normative conflict can further emerge from conflicting descriptive norms, that is, observations of actual behavior of various important social groups (Cialdini, Kallgren, & Reno, 1991). R. I. McDonald, Fielding, and Louis (2012) found that the presence of conflicting descriptive norms increased individual intentions to engage in pro-environmental behavior, if they already had strong personal predispositions toward the behav-

ior. However, individuals who had less positive attitudes saw normative conflict as a reason not to engage. Thus, depending on individual predispositions, normative conflicts can stimulate both a fight reflex towards improving group outcomes, or the opposite, desertion.

So far, all normative conflicts described have been internal, occurring within the consciousness of the individual. They may pull an individual in different directions, problematizing the choice of a course of action. The work described has also largely assumed normative conformity – that is, despite an apparent conflict, one norm has to be chosen over the other for the individual to be able to act (Oren, Luck, Miles, & Norman, 2010). However other research suggests that individuals may sometimes choose to challenge these conflicting norms.

Specifically, Packer (2007; 2010) argues that normative dissonance can result in expressions of manifest disagreement from team members. Packer's work suggests that when a group member observes an inconsistency between accepted standards of behavior (prescriptive norms), and actual activity within the group (descriptive norms), group members who identify strongly with the group would highlight this inconsistency in order to help reconciliation. This is because when a social group is part of an individual's own identity, the success of the group is closely linked with personal success (Tajfel, 1978; Turner, Hogg, Oakes, Reicher, & Wetherell, 1987). Thus individuals would be motivated to raise normative conflicts to allow the group an opportunity to resolve the contradiction. On the other hand, if an individual does not strongly identify with the group, normative conflict may lead to passive compliance or leaving the group altogether because the individual is less invested in the group's outcome (Packer, 2007).

The majority of the normative conflicts presented above have been examined in offline settings. However, as the relational view of CMC literature in Chapter 2 showed, virtual groups are just as likely to develop and evolve group norms over time.

In particular, it may be just as easy, or even more so, to observe the behavior of others in a virtual setting, and obtain a sense of group descriptive norms. Peer production systems like Wikipedia and Free and Open Source Software development rely heavily on communication technologies to document the activity of others in the team (Bolici et al., 2009). The intention is to provide a greater sense of context and transparency about how work is being done, and thereby facilitate more implicit coordination. GitHub, a hosting service that is free for FOSS projects, allows anyone to view and subscribe to either project or individual activities (Dabbish, Stuart, Tsay, & Herbsleb, 2012). This level of transparency facilities greater visibility of the descriptive norms of the group, that is, estimations of actual group behavior. At the same time, many more complex peer production systems publish a set of project guidelines and expectations for team member behavior (prescriptive or declarative group norms) (T. Ghosh, Yates, & Orlikowski, 2004). As a result, this transparency may in fact be more likely to highlight dissonance between the prescriptive and descriptive norms of the group, if they exist.

Thus members of ongoing virtual teams that have had a chance to develop salient group norms and rely heavily on computer-mediated communication may be more likely to experience normative conflicts. Chapter 1 already showed initial evidence of possible normative conflicts. To recap, Elliott and Scacchi (2003) found that normative conflicts arose when the prescriptive norms of using only Free Software were not in line with the descriptive norms of using non-free software tools to create project artwork. In the context of Wikipedia, Matei and Dobrescu (2010) found that disagreements about interpreting Wikipedia's Neutral Point of View policy were an essential element of working in the Wiki project.

Furthermore, Chapter 3 has shown that FOSS teams experience conflicts arising from incompatible or unclear group norms alongside conflicts arising from task issues, procedural challenges, interpersonal incompatibilities. While

task, process and affective conflict are commonly studied aspects of the intragroup conflict taxonomy, normative conflict is a potential extension of this taxonomy that is particularly salient for ongoing virtual teams. Findings from study 1 suggest that normative conflict has a distinctly different impact on the team compared with task, process or affective conflict. Specifically, normative conflict is a key driver of team development and change because it raises normative contradictions that result in the reformulation or clarification of group norms, as well as potential structural changes such as the emergence of faultlines. It is therefore important to quantitatively examine whether normative conflict may exist as a fourth intragroup conflict dimension:

RQ1: Do FOSS teams experience normative conflict as a distinct and separate dimension from task, process and affective conflict?

The following sections will introduce the remaining variables of interest in this study, and their relationships with each other, starting with FOSS team outcomes.

4.1.2 Outcomes

Intention to remain (ITR)

When teams are largely driven by voluntary participation, the extent to which team members intend to remain as contributors is a critical aspect of team vitality. In fact, the intention to perform a behavior has been consistently found to be the strongest predictor of subsequent behavior (Ajzen, 2011; Ajzen & Fishbein, 1977; Campion, Papper, & Medsker, 1996). However, the construct has been treated in different and somewhat contradictory ways across several decades of research in this area. Concepts such as commitment to the organization, behavioral intentions to quit and affective attachment to the organization are sometimes treated interchangeably and at other times as distinct outcomes. For example, Mowday, Steers, and Porter (1979) developed and validated a popular organizational commitment scale that contained both behavioral compo-

nents, such as intention to leave the organization, and attitudinal components, such as an emotional or intellectual bond to the organization. In recent work on open collaborative systems, Dabbish and colleagues (2012) also treat commitment as a complex construct involving affective attachment, intention to stay and exert effort on behalf of an organization. However, other studies find the behavioral and attitudinal components of organizational attachment to be distinct and complex constructs in their own right (Landau & Hammer, 1986). Studies also distinguish between commitment at the organizational level and team level (Bishop, 2000), while other work only examines one of these factors in isolation (Dayan, 2010; Jehn, 1995; Pazos, 2012).

Scholars like Michaels and Spector (1982) suggest a causal link between these variables, viewing affective commitment to an organization as a precursor for the intention to remain in a team. This perspective is consistent with Ajzen and Fishbein's (1977) classical work on behavioral intentions. Specifically, they suggest that an individual's intention to engage in a behavior is affected by an individual's attachment to a reference group and attitude toward the behavior. Thus the extent to which an individual feels a sense of belonging to the group determines intention to remain as part of that group.

Supporting this, Fang and Neufeld (2009) found that sustained participation rests on team identification alongside other factors like learned experience with the project. Similarly, Bagozzi and Dholakia (2005) have found that intention to participate in Linux User Groups depends on their identification with the group alongside other variables like developer attitudes toward working in the project.

Taken together, the body of work suggests identification with the project is strongly linked with sustained participation in open collaborative systems. The present study thus examines identification with the team in connection with developers' intention to remain in the project. The next section expands further on this relationship.

Identification (IDENT)

Identification is a classical construct rooted in Tajfel's (1978) seminal work, in which he argues that an individual's self-definition consists of not just personal characteristics, but also memberships in social groups such as racial groups, religious, political or community associations. In a later theoretical extension, Turner and colleagues (1987) argue that individuals may categorize themselves as part of a variety of groups, with different identities being more salient at different times. Thus when a group identity is more salient than an individual one, group members experience greater emotional attachment with the group, and the desire to maximize group outcomes. In other words, team goals become personal goals (Turner et al., 1987). A focus on the group further creates normative pressure on an individual to act in a certain way, that is, adopt group characteristics and behaviors. Thus identification with a group encourages members to stay as a part of that group because 1) the group's collective success becomes reflective of one's own success, and 2), because leaving would be anti-normative.

Ashforth and Mael (1989) have found that it is possible to identify with organizations and/or sub units as well, thus launching a tradition of study in literature on organizational teams. Identification also continues to be relevant in a virtual team setting. A growing body of work has shown that it is possible to categorize as part of a group both quickly, based on limited available characteristics (SIDE; Reicher et al., 1995), and to develop a group identity over time (SIP; Walther & Burgoon, 1992). In fact, a strong group identity is particularly important in virtual settings because it reduces the uncertainty of the virtual environment (Fiol & O'Connor, 2005).

In traditional organizational teams, identification was found to have a negative relationship with worker turnover (Cole & Bruch, 2006). In a study on Wikipedia, Zhu et al. (2012) showed that identification with a Wiki group encourages not only increased participation, but also focuses efforts on accomplishing group goals. Similarly, K. J. Stewart and Gosain (2006) found that in

FOSS projects, adherence to team ideology was beneficial to the team in attracting and retaining developers. Thus the following relationship is expected:

H1: Developers' identification (IDENT) will be positively related with intention to remain in the project.

Performance Perception (PERF)

Aside from team vitality measured through sustained participation, it is also important to examine team success in connection with different conflict types in FOSS settings. Chapter 1 has already elaborated on the challenges of measuring success in the FOSS setting where profits and market share metrics do not readily apply. Thus when investigating team success, it is important to conceptualize this in a meaningful way for FOSS contributors. Specifically, past research has found team performance is a multidimensional construct in FOSS involving both code quality as well as success with users, and has been been examined as the relative satisfaction of developers with their own project performance (Crowston, Howison, & Annabi, 2006). This closely aligns with the classical framework by DeLone and McLean (2003), which emphasizes both quality of software produced and its success with users. Crucially, team performance emerges as a subjective construct, because in FOSS settings success is relative to the goals set out by the project and individual developer perceptions about what constitutes success. Therefore, the present study investigates developers' perceptions of their own project performance as a function of their evaluation of factors such as software quality and success with users (McDonough, Kahn, & Barczak, 2001).

Investigating developer perceptions of performance is further appropriate because what individuals consider conflict may be subjective, while not all developers may be party to all conflict episodes. Thus it is relevant to examine the direct relationship between individual's recognition of conflict on their subsequent attitudes towards team, such as performance perception. Further-

more, team potency, or the belief in the performance of the team across multiple tasks (Guzzo, Yost, Campbell, & Shea, 1993) is consistently reflective of objective performance measures (Campion et al., 1996; Gully, Incalcaterra, Joshi, & Beauien, 2002; C. Lee, Tinsley, & Bobko, 2002; LePine, Picollo, Jackson, Matheiu, & Saul, 2008).

Specific hypotheses related to conflict dimensions and performance perception will be developed below.

4.1.3 Conflict

Chapter 2 has highlighted the way conflict within groups has been studied across many disciplines, and conceptualized in different ways. Drawing from literature on organizational teams, this study defines conflict as an interactive process manifested in incompatibility, disagreement, or dissonance within or between social entities (Rahim, 2002). Crucially, conflict involves opposing interests that are manifest in some way within the group. That is, for conflict to occur, incompatibilities must not only be present, but also visible and evaluated as conflict by one or more group members. Consequently, this study investigates developer perceptions of manifested conflict within their team. Following the previous chapter, as well as prior work by Arazy et al. (2013) and van Wendel de Joode (2004), the present study draws on literature on traditional virtual teams and distinguishes between several different forms of conflict.

Task Conflict

Past research highlights a cognitive dimension to conflict that involves differing opinions about the work that needs to be done (Jehn, 1995). Task conflict in peer production settings involves differences of opinion about what content to add and proposed changes to the project output (Arazy et al., 2013). As Chapter 3 has shown, in the case of FOSS projects, this means disagreements about proposed features and implementations. Furthermore, because many FOSS projects

are modular and depend on other projects, task conflict may also arise from differing opinions on the choice of project dependencies.

On one hand, cognitive (or task) conflict is expected to have a positive influence on team performance because it stimulates creativity through diversity of ideas and prevents groupthink (Nemeth & Goncalo, 2011). On the other hand, team members may interpret disagreements concerning the work they are doing as a negative assessment of their own abilities, causing stress and dissatisfaction (de Wit et al., 2012).

In practice, task conflict has had mixed effects on group outcomes. In classic work on traditional virtual teams, Amason (1996) found task conflict to be positively related to affective acceptance of team decisions as well as decision quality. However, Jehn (1995) found task conflict to be negatively linked with team satisfaction and team performance, while O'Neill, Allen, and Hastings (2013) found a negative relationship with potency (performance perception). A review by De Dreu and Weingart (2003) found task conflict to be negatively related to both performance and satisfaction with the team, however a later meta-analysis by de Wit et al. (2012) showed no direct effects on either.

In virtual team settings, Lira et al. (2008) found that task conflict has a negative relationship with performance perception. Similarly Windeler et al. (2015) found task conflict to have a negative impact on team effectiveness through reducing shared understanding. However, these studies were conducted on short-term virtual teams. In real world, ongoing organizational virtual teams teams, Jong et al. (2008) found task conflict to be positively connected with performance while teams with low virtuality had a negative relationship with task conflict. This is because task conflict helps to reduce uncertainty about the task at hand (Hinds & McGrath, 2006). It is possible that ongoing virtual teams have less time pressure to perform and thus are able to reap the benefits of greater cognitive diversity stimulated by task conflict.

In fact, in the context Wikipedia groups, Arazy et al. (2011) found task

conflict was also positively linked with quality of output in cognitively diverse groups. However, a later study did not find significant effects of task conflict on the quality of Wikipedia articles (Arazy et al., 2013).

Another possible reason for these mixed findings is that task conflict has varying effects on performance depending on the complexity of the task being performed (Kankanhalli et al., 2007). Specifically, task conflict was found to be beneficial for teams performing more complex tasks as it stimulates creativity and new ideas (Jehn, 1995). However, for teams working on more routine duties, task conflict may be detrimental because it takes focus away from work. As FOSS development is a creative task that involves writing complex software, task conflict may have a more positive impact on team performance. Thus the following is proposed:

H2a: Task conflict (TASK) will be positively related with performance perception in FOSS development teams.

There may also be a potential curvilinear relationship between task conflict and performance perception. By breaking task conflict further down into different levels of intensity, Qiu, Wang, and Cui (2013) found a complex, curvilinear relationship that closely echoed work on traditional virtual teams (De Dreu, 2006; Farh, Lee, & Farh, 2010). At extreme ends, task conflict appears to lead to worse output quality, with medium levels of conflict best at early stages, suggesting a kid of "Goldilocks" effect – not too little, not too much, and at the right time (De Dreu, 2006; Farh et al., 2010; Qiu et al., 2013). Thus a related hypothesis is proposed:

H2b: Moderate levels of task conflict (TASK) will be more positively related with performance perception in FOSS development teams than high or low task conflict levels.

Affective Conflict (AFFECT)

Affective (or relational) conflict involves emotional disagreements and interpersonal dissonance, thus it is often presented in contrast to task conflict (Jehn, 1995). These displays of emotion and personal incompatibilities take focus and resources away from accomplishing the group goals, and are therefore viewed as counter productive to team outcomes and member well-being (Jehn, 1995; Mortensen & Hinds, 2001). In virtual communities like FOSS projects, affective conflict may further manifest as disinhibited interactions between members that involve personal attacks, such as flaming (Arazy et al., 2013; Franco et al., 1995). At the same time, as the review in Chapter 2 highlights, in the absence of situational or contextual cues, FOSS contributors may attribute failure to personal incompetence or inattention, rather than situational factors such as lack of information. This may result in a more pronounced effect of affective conflict on the team.

Supporting this, in both collocated and virtual teams, the overall effect of affective conflict has been found to be predominantly negative for team performance (Jehn, 1995, 1997; Jehn & Mannix, 2001; Mortensen & Hinds, 2001; de Wit et al., 2012). Affective conflict was also found to have a negative relationship with perceptions of team performance in both collocated (O'Neill et al., 2013) and virtual teams, with virtual teams experiencing greater negative effects (Lira et al., 2008). The direct impact of affective conflict has not been empirically investigated in research on peer production. However, work by Arazy et al. (2013) on the interaction between task and affective conflict suggests affective conflict, when present, has a negative impact on the quality of Wikipedia articles. Taken together, the literature suggests affective conflict can be expected to have a negative relationship with perceived performance because it detracts from the task at hand. Thus the following hypothesis is proposed:

H3a: Affective conflict (AFFECT) will be negatively related with perception of project performance

Affective conflict also has consistently negative relationships with team member well-being markers such as satisfaction (De Dreu & Weingart, 2003), cohesiveness (Jehn et al., 2008), and team viability (Tekleab et al., 2009). The present work argues that affective conflict will impact team viability by reducing individual identification with the team by focusing on individual dissimilarities rather than shared group similarities, and therefore undermine individual intentions to remain in the project. Thus the following is proposed:

H3b: Affective conflict (AFFECT) will be negatively related with team identification.

Process Conflict

Process conflict is a third conceptual dimension that involves disagreements over how to perform a task and team responsibilities. Process conflict has appeared less frequently in research due to measurement difficulties in distinguishing it from the first two conflict types (de Wit et al., 2012). However, Behfar and colleagues (2011) have recently re-examined this dimension and highlighted its distinctiveness in predicting team outcomes.

Specifically, disagreements about procedure introduce confusion about the correct course of action, and take attention and resources away from the task at hand. Therefore, in the context of traditional virtual teams, process conflict has had largely negative effects on team performance in traditional virtual teams (de Wit et al., 2012). Furthermore, when comparing teams with both high and low virtualness, Jong et al. (2008) found process conflict to have a consistently negative impact on performance across all teams.

The direct effect of process conflict on outcomes has not been examined in peer production settings as well. However, Arazy et al. (2013) have found that in Wikipedia, the presence of process conflict alongside task conflict has a negative effect on article quality. We can also expect process conflict to be a relevant and negative force in FOSS team performance. FOSS teams evolve and

infer procedures based on visible indirect cues and the behavior of others (Bolici et al., 2009). As Chapter 3 highlights, this lack of explicit coordination may lead to instances of process conflict through misunderstandings about issues such as access to the code base, as well as how to structure contributions in terms of an expected coding style. These issues can take attention away from the task at hand and reduce perceptions of team performance.

This would be especially true for ongoing teams that place particular emphasis on the development of efficient processes (Saunders & Ahuja, 2006), and would thus dedicate attention and resources away from the task at hand toward resolving the procedural issue. Process conflict would also be especially relevant in the virtual setting. Virtual teams are particularly vulnerable to uncertainty (Hinds & McGrath, 2006) thus process conflict may highlight a lack of clarity about team procedures. Therefore, we can expect high levels of process conflict to negatively impact perception of project performance:

H4: Process conflict (PROC) will be negatively related with perception of project performance.

Normative Conflict

Lastly, Chapter 3 has proposed a fourth conflict dimension for more long-term oriented, self-organizing teams that have had time to develop group norms, such as FOSS projects. An earlier section has already described its potential significance to the intragroup conflict taxonomy. To recap, normative conflict, or conflict about group norms, involves higher order disagreements about group function that do not directly arise out of working together. Instead, conflict emerges from a perceived dissonance between the prescriptive norms of the group, that is what the group aims to do, and the descriptive norms, or members' actual behavior (Packer, 2007). Thus normative conflict may involve ideological debates and recursive, meta-level, discussions on the state of the community as a whole. Though outlined qualitatively in Chapter 3, this form of conflict has not yet been

empirically examined in the context of ongoing virtual teams.

Similar to affective and procedural conflict, normative conflict may take attention away from the activity of writing software while the community debates larger issues, thus negatively impacting perceptions of performance. In line with Coser's (1957) predictions, Chapter 3 has highlighted how normative issues such as the Node.JS gendered commit debate become the focus of community discussions because issues about group norms and values can potentially affect every member of the group. Therefore, the following hypothesis is proposed:

H5a: Normative conflict (NORM) will be negatively related with performance perception

Additionally, because normative conflict highlights inconsistencies in the mission and values of the project, it may lead to perceptions that group cohesion is overall lower, and thus reduce identification with the project. Specifically, normative conflict may highlight factions of opinions on the same issue that highlight fundamentally different values. This may reduce overall group identification, and promote identification with sub-groups instead, creating faultlines (Lau & Murnighan, 2005). This may be particularly relevant to virtual teams because, in absence of other visually identifying cues, different factions of opinions may become the primary salient social categories, as the SIDE theory suggests (Reicher et al., 1995). Therefore, the following additional hypothesis is proposed:

H5b: Normative conflict (NORM) will be negatively related with team identification

Given the potential relationship between developer identification with the team and intention to remain in the project, as well as the fact that affective, and normative conflict are predicted to impact identification, it would be pertinent to examine their exact mechanism of action. Specifically, this study will also explore whether affective and normative conflict reduce intention to remain indirectly by reducing team identification. Therefore the following additional

hypotheses are proposed:

H5c: Identification will mediate the relationship between affective conflict and intention to remain.

H5d: Identification will mediate the relationship between normative conflict and intention to remain.

Interactions

Finally, a number of recent studies suggest that these conflict dimensions do not occur in isolation, and influence the effects of other conflict types on outcomes (Arazy et al., 2013; de Wit et al., 2012). In traditional virtual teams, de Wit et al. (2012) have found that in the presence of affective conflict, task conflict and performance are less positively associated. This may be because, as the previous chapter has shown and a growing number of studies suggest, task related criticisms are often misinterpreted as a personal attacks (Jehn & Mannix, 2001; Mooney, Holahan, & Amason, 2007; Simons & Peterson, 2000). Furthermore, while individuals attribute their own mistakes to situational factors (e.g. I couldn't reply to an e-mail because my service was offline), they often attribute others' mistakes to personality faults (e.g. They didn't reply my e-mail because they are irresponsible). This may lead to the evolution of initially task related conflicts into affective disagreements.

In addition to this, Arazy et al. (2013) found in the Wikipedia setting that task conflict could evolve not only into affective but also process conflict. This may be because manifest task conflict may raise uncertainty about the task at hand. If this remains unresolved, this may lead to a variety of conflicting procedures designed towards approaching different aspects of the task. The authors also found both task-affective and task-process conflicts to have a negative relationship with Wikipedia article quality (Arazy et al., 2013).

In addition to this, Greer et al. (2008) have argued that all three conflict types (task, affective and process) can evolve into each other over time under different

conditions. Supporting this, they found that high levels of initial process conflict were connected with high levels of affective, task and even more process conflict across future time periods. They also found the three conflict types interacting at team inception, suggesting that the same events may evolve into any one of the three conflicts over time. This recursive cycle may arise when team members misinterpret other's behavior in response to their own expression, and respond to the misinterpretation in a more hostile way, exacerbating the issue (Thomas & Schmidt, 1976).

Finally, qualitative work in the previous chapter suggests that task, affective or process conflicts may also evolve into normative conflict. For instance, task conflict can take on a normative dimension when manifested task uncertainty uncovers deeper values inconsistencies, such as the conflict episode Elliott and Scacchi (2003) describe concerning using non-free tools to create images for a Free software project. Process conflict can similarly evolve into normative issues, for instance, when the Node.JS community faced a procedural issue due to rejecting a trivial commit that raised bigger issues about how to discuss gender in the community (Chapter 3). Finally affective conflict can further evolve into normative conflict, such as a "transphobic" comment in the Opal project that sparked a discussion about freedom of speech and respecting diversity in the community (Chapter 3). As there appear to be a number of potential interactions, and the interaction of normative conflict with other conflict types has not yet been examined, a more general research question is proposed:

RQ2: How do interactions between task, affective, process and normative conflict influence perceptions of team performance and identification with the team?

4.1.4 Conflict antecedents

The rich history of inquiry into intragroup conflict identified a number of relevant antecedent factors both at an individual and group level (de Wit et al.,

2012). As this study is interested in manifested intragroup conflict, group level antecedents are considered as main variables in the model. Specifically, this study examines four relevant conflict antecedents suggested in Chapter 3: task interdependence, geographical distribution, leadership style and the relative distribution of decision-making, alongside several control variables.

Task Interdependence

Interdependence is a fundamental element that makes a group of individuals a team because it spurs cooperation through shared tasks, goals and incentives (Deutsch, 1949). While there are many kinds of interdependence, task interdependence, that is, a condition in which individual outputs are influenced by the actions of others, is the most commonly studied and therefore the focus in this study (Wageman, 1995).

Despite the advantages of positive interdependence in promoting cooperation, it may also lead to greater conflict levels (LePine et al., 2008). For instance, Saavedra, Earley, and Van Dyne (1993) find that task interdependence has a positive relationship with overall conflict levels in a team. Research has also examined the impact of interdependence on individual conflict types. Specifically, Jehn (1995) finds that greater interdependence involves a greater need to exchange opinions concerning the task at hand, and therefore significantly increases task conflict levels.

Furthermore, interdependence significantly increases affective conflict levels because individual incompatibilities are more likely to come to the surface as a result of the greater need to work closely together (Jehn, 1995). We may similarly expect task interdependence to increase process conflict levels. Greater task interdependence requires more complex rules, access and authority structures and therefore affords greater opportunity for process conflict Kankanhalli and colleagues (2007) find that greater task interdependence will lead to more affective conflict and thus indirectly reduce team performance virtual team set-

tings. However, in contrast to this, Stark et al. (2014) found that interdependence was negatively related to both affective and process conflict in virtual teams. Given the somewhat contradictory findings, and that most research on the relationship between conflict and interdependence in virtual team settings involved short-term student teams, it would be interesting to examine this relationship in an ongoing virtual team setting like FOSS development.

Even though FOSS developers do not often explicitly collaborate on tasks and prefer to work in modular individual chunks (Bolici et al., 2009), they are interdependent in other ways. Specifically, research also identifies 1) pooled interdependence in which smaller tasks are performed individually and pooled together, 2) sequential interdependence in which different stages of the task are performed individually by different team members, and 3) reciprocal interdependence whereby individuals take turns performing tasks (for instance, paired programming) (Saavedra et al., 1993). Instead of directly collaborating with others on a task, FOSS teams instead employ pooled and sequential task interdependence, that is their individual efforts are pooled into (usually) one code base, with some projects also involving a gatekeeper (such as a maintainer) who gives feedback and approves the code contribution.

Thus task interdependence continues to be a relevant concept to examine in FOSS teams. In fact, qualitative work by van Wendel de Joode (2004) suggests that more modular FOSS projects will experience less overall conflict than more interdependent ones. In the context of Wikipedia, Kittur and colleagues (2010) have also found that contributor density, or the degree to which editors are forced to interact on an article, increases conflict levels.

Taken together, we can expect that in ongoing virtual teams like FOSS teams, task interdependence will be positively connected with conflict occurrence. Specifically, task interdependence creates more opportunities for opinions to diverge, leading to more task conflict. At the same time, greater task interdependence also requires more complex rules, access and authority structures

and therefore greater opportunity for process conflict. Finally, interdependent teams may experience the effects of affective conflict more severely than more modular team structures where individuals can more easily avoid confrontation. Therefore the following relationship is proposed:

H6a: Interdependence (INTER) will be positively related with the occurrence of task, affective, and process conflict.

Interdependence may also have a direct impact on team emergent states. Specifically, LePine and colleagues (2008) find in their meta-analysis of work on interdependence that it is consistently connected with greater identification in the team. Taken together with the above proposition, conflict may mediate the positive relationship between interdependence and team identification. Therefore, the present study also explores the following hypothesis:

H6b: Conflict will mediate the positive relationship of interdependence and identification with the team

Geographical Distribution

Chapter 2 has already introduced the significance of geographical distribution to virtual team development and conflict. Specifically, the chapter highlighted that not all virtual teams are necessarily geographically dispersed groups because teams in the same location are also increasingly using computer-mediated communication to collaborate. Research also showed that when controlling for team longevity, virtualness alone did not predict conflict levels (Jong et al., 2008). Thus it is not the virtualness itself, but the distribution of team members across locations, coupled with time zone dispersion and cultural differences, that are likely to impact conflict occurrence in ongoing virtual teams.

Though there may be exceptions, such as organizational collocated teams working on software that is later released as open, FOSS development is on the whole highly geographically distributed.

Demographics surveys consistently find that a majority of FOSS contribu-

tors come from the US and countries within Europe, such as the Netherlands, Spain, France, and Italy among the top 10 (Arjona-Reina et al., 2014; Robles et al., 2001). However, even projects distributed between the US and Europe are separated by many time zones, and have very different communication norms that may promote misunderstanding. For instance, the United Sates and the Netherlands often have very direct communication styles, which may conflict with that of Spain, France and Italy who rely more on non-verbal cues and indirect messages to communicate (Hall, 1976). Furthermore, there is an increasing participation of non-native English speakers, and communities that are non-natively English by default, such as the Ruby community, who communicate a big portion of coordination messages in Japanese. In text-based environments such as those used by FOSS projects (Slack, GitHub, Mailing lists, IRC) translation and lack of fluency in language may further present miscommunication opportunities.

Geographical distribution may result in a combination of cognitive, time, or cultural diversity. Cognitive diversity leads to greater task conflict by bringing together more diverse perspectives and ways of doing things (Olson, Parayitam, & Yongjian Bao, 2007). Furthermore, meeting in person is challenging for highly geographically distributed teams, and continual reliance on computer-mediated communication may result in depersonalization of other team members and more disinhibited communication, leading to greater affective conflict (Mortensen & Hinds, 2001). Culturally diverse teams are further likely to misinterpret task or procedural issues as personal attacks, thus also resulting in greater affective conflict (Mooney et al., 2007). While distance may make interpersonal incompatibilities easier to avoid when developers do not physically interact with each other, Chapter 3 has shown that a lack of responsiveness due to more independent teamwork may be interpreted as a personal affront by team members, leading to more affective conflict. At the same time, time zone dispersion complicates coordination and increases communication overhead (Cramton, 2001).

Finally, as a whole, more distributed teams have less shared context, and therefore experience greater overhead when coordinating (Hinds & Bailey, 2003). This can lead to more misunderstandings about procedure and thus greater process conflict. Thus the following hypothesis is proposed:

H7: Geographically distributed (DISTRIB) teams will experience more task, procedural and relational conflict

Leadership Style

Leadership one of the most established domains of inquiry in studies on organizations. Similar to work on interdependence, it would not be possible to do justice to this body of work in a small sub-section. Thus this section is focused on one of the more popular characteristics of effective supervisors, that of transformational leadership (Bass, 1985).

Transformational leadership is often contrasted with a transactional leadership style: while transactive leaders rely on rewards and punishments to manage and motivate their team, transformational leaders are charismatic, display strong commitment to ideals, inspire and lead by example (Judge & Piccolo, 2004). Literature and reviews consistently find that transformational leadership enhances performance in a wide range of organizational settings (e.g. Judge & Piccolo, 2004; G. L. Stewart, 2006).

Research also finds that transformational leadership is particularly relevant in the virtual team context. For instance, Bell and Kozlowski (2002) argue that because of more limited communication channels, it takes more effort to be a proactive leader in distributed teams. Thus a transformational leadership style would be more efficient. At the same time, Avolio and Kahai (2003) find that the effort of transformational leadership may be enhanced in conditions of limited social cues. Transformational leadership often focuses on emphasizing work directed at the benefit of the group or organization as a whole, while conditions with limited social cues reinforce this by increasing the salience of a collective

identity (Avolio & Kahai, 2003). Furthermore, Purvanova and Bono (2009) have found that transformational leaders have a greater impact on positive team outcomes when there is more uncertainty, such as in virtual rather than face-to-face settings.

However, Hambley, O'Neill, and Kline (2007) found no difference in the effects of transactional and transformational leadership in text based, video conferencing or face-to-face conditions in virtual teams. This may be because the study employed short-term student teams who did not have time to develop a more stable and long-term collective identity that transformational leadership supports. These conflicting findings warrant further investigation in an ongoing virtual team setting.

In an adaptation of adaptive structuration theory, Avolio, Kahai, and Dodge (2000) argue that over time, leadership may co-evolve with introduction of technology just as it may guide actions in virtual teams about the appropriation of this technology. Similarly, in the FOSS context O'Mahony and Ferraro (2007) have shown governance to be an emergent property, evolving as a result of different structural, technical and social conditions. Thus we can expect to see variation in leadership styles across FOSS projects.

Transformational leadership has an important effect on conflict in distributed work settings. Dionne, Yammarino, Atwater, and Spangler (2004) argue that leadership promotes more functional and less dysfunctional conflict. Kotlyar and Karakowsky (2006) showed that transformational leadership lowers cognitive conflict. Supporting this, Wakefield, Leidner, and Garrison (2008) have found that effective leaders mitigate task conflict by acting as monitor, and process conflict through coordination activities. Additionally, transformational leaders may mitigate affective conflict levels by acting as mediators (van Wendel de Joode, 2004) and normative conflict levels through charisma and inspiring developers to align towards group goals. Hence, the following is proposed:

H8a: Transformational leadership (LEADER) will be negatively associated

with all types of conflict

Furthermore, leadership may have a direct effect on performance perception and intention to remain through team identification. For instance, Giuri et al. (2008) have found that leadership is an important part of FOSS project success, while Li, Tan, and Teo (2012) have shown that transformational leadership is an important factor in motivating FOSS contributions. Thus the following additional hypothesis is proposed:

H8b: Conflict will mediate the positive relationship between transformational leadership, performance perception and team identification

Distribution of Decision-Making

Leadership research is also concerned with the relationship between leaders and subordinates, which is often a complementary level of inquiry to leader characteristics described above. One of the areas that has attracted attention from research is that of shared leadership (or team empowerment). Though the concept in organizational teams is not a new one, it has recently gained more attention because of the increasingly self-determining nature of virtual teamwork (Tannenbaum et al., 2012).

As early as 1954, Gibb remarked that "leadership is probably best conceived as a group quality, as a set of functions which must be carried out by the group" (Gibb, 1954, p. 54). Conger and Kanungo (1988) argued that while power was a critical aspect of the dynamics between team members, it need not be concentrated with team leaders. They put forward the concept of team empowerment to represent conditions when leader power is shared with subordinates. It is important note that shared leadership does not necessarily mean the absence of a leader entirely, rather a downward redistribution of power (C. L. Pearce, 2006). Thus it is relevant to examine the extent to which leaders exhibit transformative characteristics, alongside the level of empowerment, or extent of distribution of decision-making, in the team.

Shared leadership is particularly relevant in conditions of high creativity and high task complexity that are prevalent in FOSS development (J. P. Pearce, 1993). When tasks are more complex, such as developing a large code base, it is less likely that one person can be an expert at every aspect. Thus teams may tend toward taking more responsibility for individual chunks. Software development is also a highly creative activity, especially in communities that value writing simple, clean and beautiful code that represents complex ideas.

Shared leadership is also more appropriate for unstructured tasks (Avolio & Kahai, 2003), a common feature of FOSS development. Tasks performed by FOSS developers are often self-selected based on more broad project aims and feature plans (Puranam et al., 2014), and their execution is largely left to the individual developer who proposes possible solutions and discusses them with peers and core team members (Chapter 3). Finally, shared leadership is also particularly relevant under high team heterogeneity because it encourages team reflection that in turn leads to more cohesion and positive outputs (Somech, 2006).

Bell and Kozlowski (2002) argue that virtual teams are particularly predisposed to a shared leadership structure due to their reliance on computer-mediated communication and distributed nature that make keeping an overview of all aspects of team function more challenging. Thus virtual team leaders are more likely to create structures and processes in place of leadership activities, as well as to distribute their power toward subordinates. Supporting this, Hoch and Kozlowski (2014) find that hierarchical leadership is related to better performance only in low virtuality conditions, but not in highly virtual teams. However, they also find that shared team leadership is positively related to performance in both virtual and collocated teams. In fact, shared leadership has been consistently linked to improved team performance in collocated teams as well (Carson, Tesluk, & Marrone, 2007). For instance, Campion and colleagues (1996) have shown that a more participatory decision making style is an impor-

tant predictor of organizational team success.

Surprisingly, relatively little work has examined the impact of shared leadership alongside conflict emergence. In one exception, Acar (2010) found that shared leadership reduces affective conflict by moderating the effects of diversity. Furthermore, participatory decision-making can be expected to reduce process conflict levels as team members gain more autonomy and contribute more to team organization. Similarly, there may be less need for normative conflict in more participatory teams because members could have more formal opportunities to vote on and impact the growth and development of the team. Thus the following is proposed:

H9a: Participatory decision-making (DECIS) will be negatively related with affective, process and normative conflict

Finally, the above section has also shown that team empowerment can be directly connected to improved team outcomes. However, Chen, Sharma, Edinger, Shapiro, and Farh (2011) found that when teams experience more conflict, team empowerment has less positive effects on team outcomes. Thus conflict may be expected to reduce the positive relationship between shared leadership and identification with the team:

H9b: Conflict will mediate the positive relationship between participatory decision-making (DECIS) and team identification

4.1.5 Control variables

Though the study focuses on group-level antecedents and the manifestation of conflict at the group level in the model, there may be other individual- and group-level factors that influence outcomes. Specifically, the study also controls for gender because research on effects of gender diversity on peer production team outcomes have received mixed results. On one hand, Collier and Bear (2012) have shown that highly contentious peer production environments may lead to lower participation intentions for female contributors. By contrast,

Vasilescu, Filkov, and Serebrenik (2015) have found that that gender diversity does not have a significant relationship with turnover. Furthermore, length of contribution is also examined as a control variable because FOSS teams are ongoing teams, and individuals who have participated longer may be more likely to identify strongly with the team and have greater intention to continue contributing to the project (Saunders & Ahuja, 2006). In fact, recent work by Vasilescu et al. (2015) has shown that both greater gender and tenure diversity significantly predict greater productivity. Similarly, developers with more central roles in the project may perceive themselves as more prototypical members of the in-group, and have stronger identification and intention to remain in the project (Turner et al., 1987). Thus developers' roles in the project are also examined as a control. Finally, as larger projects tend to be more complex, they may experience greater negative effects of conflict (Amason & Sapienza, 1997), thus project size is further controlled for.

The next chapter describes the operationalization of the above propositions, presents the methodology for the second study, as well as the relevant findings. Findings relevant to the specific aims of the second study are discussed at the end of Chapter 5. Chapter 6 discusses findings from both studies in greater detail, relates them to broader dissertation aims and discusses their implications for theory and practice.

Chapter 5

Study 2 - Survey

As Chapter 4 has introduced, Study 2 aims to address the following research aims:

1. Validate the four-factor structure uncovered by Study 1 and, in particular, examine the relevance of normative conflict in ongoing team settings.
2. Use this updated framework to investigate the kinds of conflict that affect developers' attitudes towards their team and their sustained participation;
3. Examine different structural input variables to understand their impact on conflict emergence.

To address these aims, Chapter 4 has outlined a conflict model that involves four conflict dimensions (task, process, affective and normative conflict), structural inputs (interdependence, distribution of decision-making, leadership style and geographical distribution), team emergent states (identification and performance perception), as well as outcomes (intention to remain in the team). The present chapter describes the methodology used in Study 2 to test the proposed model and the effects proposed in Chapter 4, as well as the study findings. The chapter concludes with a discussion of the findings as they relate to the study aims, as well as extant literature on the FOSS phenomenon and virtual teams more broadly.

5.1 Methodology

5.1.1 Measures

Most of the measures used in the study were adapted for the FOSS context from existing pre-validated scales used in traditional organizational settings, with the exception of normative conflict which was designed based on the results of the qualitative work described in Chapter 3. All items were measured on a 5-point Likert Scale (from Strongly Disagree to Strongly Agree), and their operationalization is described below. Throughout the survey, respondents were instructed to answer the questions from the perspective of the project they participate most actively in at the moment – this is referred to as "this project" in the questionnaire items. The following section describes the operationalization of the constructs used in Study 2, beginning with the emergent states, outcome and input measures. Two rounds of pilots inform the design of the final instrument, with the complete survey questionnaire employed in the final study available in Appendix .3.

Antecedents, emergent states and outcomes

The scale for task interdependence was adapted from work by Yetton and Sharma (2007), and included 6 items such as "My work requires frequent coordination with the effort of others" and "My work can be performed fairly independently of others" (Reverse coded).

Geographical distribution was adapted from Chudoba et al. (2005), and included 4 items, such as "I collaborate with people in different time zones", "I collaborate with people who speak different native language or dialects than mine" and "I collaborate with people I have never met face to face".

Transformational leadership is usually measured using the Multi-factor Leadership Questionnaire (Avolio, Bass, & Jung, 1999), which contains around 80 items. However, to prevent response fatigue, the short transformational lead-

ership scale validated by Carless, Wearing, and Mann (2000) was used instead. The scale uses one item to represent each of the dimensions of transformational leadership, and therefore measures the overall level of transformational leadership present in the project. Items include "The leader/founder of this project gives encouragement and recognition to contributors" and "The leader/founder of this project communicates a clear and positive vision of the future for this project". Participants who identified as project leaders at the beginning of the survey were excluded from answering this question.

Participatory decision-making was adapted from Campion et al. (1996) and included 6 items such as "The project is designed to let everyone participate in decision-making" and "As a contributor to this project, I have a real say in how the group carries out its' activities".

The measure for identification was adapted from Greene (2014), and included 6 items such as "This project's successes are my successes" and "I have a number of qualities typical of members of this project".

Performance perception was measured as a feature of developers' satisfaction with their project output, that is, the extent to which the project performance matched developers' expectations within the past year. The scale was adapted from McDonough et al. (2001) using insights from S. T. Lee et al. (2009) and Crowston, Kangning Wei, et al. (2006) that suggest including both software quality and user success dimensions. The scale was anchored by the statement "How well has this project met your expectations over the past year in", and consistent of 6 items such as "Developing features that are successful with users" and "Developing high quality features". This scale used a 5-point Likert scale as well, with anchors ranging from 1 (fell below expectations) to 5 (surpassed expectations).

Intention to remain was adapted from Jehn's work (1995) and consisted of 3 items, such as "If I have my own way, I will continue working on this project" and "I have thought seriously about leaving this project" (reverse coded). In-

tent to remain appeared mid-way through the survey instrument and also served as an attention check – responses with the same score for all three items before re-coding (e.g. rating high on questions about continuing to work on this project, and also high on questions about leaving this project) were excluded from further analysis.

Conflict Measures

Measures for task, affective and process conflict were adapted from Jehn and colleagues' work (1995; 1999). The affective conflict scale was used as is. Task and process conflict measures were updated with several new items based on the qualitative work described in Chapter 3 that reflected more precisely the experience of task and process conflict for FOSS teams. In the pilot, some of the original conflict measures were combined with the new measures.

While several measures exist for normative conflict (R. I. McDonald et al., 2012; Packer & Chasteen, 2010), they are designed to measure internal and individual level dissonance, rather than the outward group-level conflict manifestations this study is interested in. As no measures exist for normative conflict in this context, new measures were developed based on the findings presented in Chapter 3.

In doing so, the existing conflict dimensions were reexamined and refrained in the context of FOSS. Thus, before implementing the final survey, two pilot tests were conducted to verify the four-factor structure of the conflict taxonomy. Details of the pilot study are described shortly below.

Controls

Gender was measured following the operationalization used in the recent FOSS survey by Arjona-Reina and colleagues (2014), by asking respondents which of the following options they best identified with: "Male", "Female" or "Other, please specify". In this way the study captured developers' own perception and

representation of their gender orientation rather than psychical characteristics.

Respondents were asked to estimate the size of their project using the following options for number of developers: “I am the only contributor”, “2-10”, “11-20”, “21-30”, “31-40”, “41-50”, “More than 50” and “I’m not sure”. Respondents who were the only contributor to their project were not asked to answer group level questions about conflict and antecedents.

The respondents’ role in the project was measured based on Crowston and colleagues’ (2006) work on the layers of participation in FOSS communities. This variable was categorical, with the following options presented: “Leader/-founder of the project”, “Core developer/maintainer”, “A member with commit access”, “I do not have an official title”, “Others, please specify”.

Length of respondents’ contribution to the project was measured using one item with the following options: “1 year or less”, “About two years”, “About 3 years”, “4 or more years”. For subsequent analysis, this was treated as an interval variable.

Finally, activity level in the project was measured by asking participants to specify, in hours per week, the amount of time they typically spent working on the project they are most active in.

Participants were also asked to indicate the country they were from to understand demographics of the sample, however this categorical variable was not used in analysis.

Levels of Measurement

It is important to note at this juncture that the survey measures employed were used consistently on an individual level. This was at times in contrast with the traditional use of these measures, particularly the conflict scales, whereby responses of individuals in teams were aggregated to represent a consensus variable. Specifically, in prior work, responses of the team at the individual level

were used to establish a higher order construct representing the whole team[1]. There were several reasons for this methodological diversion.

Firstly, Study 1 has highlighted the very subjective nature of some of the conflict episodes. Therefore issues that one developer may perceive as conflictual may not necessarily be perceived as conflict by another group member. If these perceptions were aggregated, they may risk under-reporting conflict levels.

Secondly, the present study was interested in understanding effects on turnover, operationalized as the intention to remain as a contributor to a given project. Intentions are highly individual constructs, thus it is logical that developer individual perceptions would be connected with developer individual outcomes. As a result, other outcome measures (performance perception and identification) were also measured at the individual level. Similarly, antecedents may also differ on an individual basis. For example, transformational leaders are considered as such when they are perceived to embody transformational leadership characteristics. Individuals within a project may have differing interactions with the same leader at different points in time, thus their perceptions may vary. Furthermore, perceptions of geographical distribution and interdependence may vary across a project depending on what area a developer is engaged in, and where other contributors who are engaged in this area are located.

Thirdly, because FOSS projects are not as tightly coupled as traditional organizational virtual teams, and "team" boundaries may change as developers move within the project to work on different things or engage in several simultaneous activities, it would be inappropriate to aggregate their perspectives as these may enforce a static structure on a group that is more fluid by default.

As a consequence of this methodological decision, the present study can only speak to the experiences, perceptions and intentions of individual developers, rather whole projects as a collective entity. To ensure that reliability and validity of the constructs was not lost through the decision to individualize

[1]For a very good discussion of different types of aggregation and their implications, please refer to work by Chan (1998)

certain aggregate constructs, two rounds of pilot tests were conducted before running the final study, and they are described in the following section.

5.1.2 Pilot tests

The first pilot study was conducted with 7 FOSS developers. Developers were invited from a pool of contacts made during the first study reported in Chapter 3 who have expressed explicit interest to participate in future research projects. The first pilot invited qualitative feedback on the extent to which the questions reflected developers' experience of working in a FOSS team. Participants were given an initial online version of the questionnaire, and a comment box on every page of the survey allowing for qualitative feedback. Participants were asked to reflect on both technical issues, such as flow, as well as the content of the questions, in particular the extent to which question wording accurately captured their experience. Overall feedback was positive and suggested the survey, and in particular item wording, handled relevant issues. Adjustments to the survey instrument were made based on feedback, clarifying instructions, and applying suggested improvements to the phrasing of questions. Next, a larger pilot study was conducted to identify measurement issues. Table 5.2 shows the items used in the second pilot test.

Questionnaire links were distributed on social media and during the annual FOSSAsia Conference held in Singapore in March 2015. In total, 58 responses to the second pilot were received, with 25 valid for further analysis after attention and quality checks.

Table 5.1 presents reliability statistics for the second pilot study. Overall, the inter-item reliability was good for most of the variables of interest, and above the traditionally accepted threshold of 0.7 (Nunnally, 1978). An exception was geographical distribution with an alpha of 0.66, the reliability of which did not improve through removal of items. As the scale was validated in previous work, and Nunnally (1978) suggests that it is acceptable in early research stages to

work with lower reliability values, the measure was retained for the final survey.

As Table 5.1 shows, all conflict variables have reliability greater than 0.8 and well above the traditional 0.7 cut-off. An exploratory factor analysis employing Varimax rotation suggested an expected 4-factor structure that explained 77% of the total variance. However, some of the items did not load cleanly on only one factor as Table 5.2 shows. As a result, several items were dropped from the final survey. Specifically, as Jehn's (1995; 1999) original items were validated across numerous studies, they were retained. In a few instances, these items cross-loaded on other factors. However, this may be the result of a lack of clarity in the newly developed items rather than a problem with Jehn's original scale. Additionally, some of the better performing new items were retained. For normative conflict, only two items performed well. These were retained, together with two other items that matched most closely the conceptual definition of normative conflict presented in Chapter 3. Further items were developed for use in the final survey that will be discussed in the following section.

Table 5.1: Means, standard deviations and reliability statistics for pilot 2

Variable (N=23-25)	α	**Mean**	**Standard Deviation**
Interdependence	.79	2.98	0.80
Geographical Distribution	.66	4.63	0.58
Leadership	.92	3.83	0.96
Decision-making Distrib	.83	3.73	0.87
Identification	.74	3.88	0.67
Performance	.84	3.64	0.74
Intent to Remain	.73	3.91	1.22
Affective Conflict	.92	2.27	1.02
Task Conflict	.90	2.90	0.80
Process Conflict	.85	2.27	0.89
Normative Conflict	.84	2.05	0.91

Table 5.2: Exploratory Factor Analysis using Pilot Data

Items	AFFECT	PROC	TASK	NORM
There is friction among contributors	**.67**	.25	.14	.56
Personality conflicts are evident	**.84**	.14	.15	.16
There is interpersonal tension among contributors	**.89**	.15	.20	.20
There are emotional displays (i.e. swearing, flaming, capitalization) among contributors	**.86**	.04	.33	.11
Contributors have different opinions regarding the work being done.	.13	.06	**.87**	.29
There is conflict among contributors about the work being done	.18	**.59**	.52	.31
There are differences of opinion about what features should be added.	.25	.35	**.75**	.18
Contributors disagree on proposed changes to this project's code base.	.51	.13	**.75**	-.18
Contributors disagree about the future direction of the code base.	.11	.57	**.58**	-.04
Contributors disagree about coding style.	.40	**.67**	.21	-.40
There are disagreements about who should do what in this project.	.20	**.89**	.16	.09
There is conflict among contributors about task responsibilities.	.29	**.75**	.28	-.04
Contributors disagree about resource allocation.	.15	**.80**	.16	.29
There are debates about the best way to perform a task.	**.57**	.35	.21	.28
Contributors to this project disagree about procedure.	**.69**	.37	.13	.14
Contributors to this project disagree about the project mission.	.31	.00	.22	**.77**
There are disagreements concerning the values this project upholds.	.17	.18	.08	**.90**
Contributors debate the agreed upon customs and norms of this project.	**.51**	.46	.34	.17
There are conflicts concerning the way this project's community functions.	**.72**	.54	.09	.03
Contributors disagree about the tone and style of communication used within the group.	**.74**	.45	.10	.13
Contributors disagree about the overall direction of this project.	.42	**.67**	-.17	.39

Extraction Method: Principal Component Analysis. Rotation Method: Varimax with Kaiser Normalization.
a. Rotation converged in 7 iterations.

5.1.3 Survey

Measures

For the final survey, all of the outcome and antecedent measures were retained as they were used in the pilot, together with measures for control variables. To improve the reliability and validity of the conflict dimensions, all of Jehn and colleagues' (1999; 1995) original scale items for affective, task and process conflict were used in the final survey. Newly designed items from the pilot were included when they had performed well together with Jehn's variables. Additionally, three more normative conflict measures were designed to refine and improve the scale and included alongside the 4 measures retained from the pilot. The retained and newly developed items are presented in a subsequent section.

Sample

The final survey randomly sampled developers on GitHub who are involved in public projects containing two or more developers. Although GitHub hosted projects are not necessarily Free and Open Source Software (FOSS), GitHub encourages the use of a FOSS license when creating a new project, especially a public one. Thus, in absence of a central repository for all FOSS projects, GitHub offers a close approximation. In addition, invitations and survey instructions specifically highlighted the target audience to be FOSS contributors.

The April 2015 version of the GHTorrent dataset (Gousios, 2013) was used to obtain the list of participants. First, project sizes were determined by counting the number of users who had commit access to the repository for each project in the dataset and constructing a list of projects with two or more contributors. Next, all project members who had a commit merged in the past year (April 2014 or later) were marked as active project contributors. Finally, to ensure diversity of projects, stratified random sampling was used to obtain an equal number of

unique active contributors from projects of various sizes: 2-10, 21-30, 31-40, 41-50, 51-100, and more than 100. Members from larger projects were over-sampled because there are considerably more small projects on GitHub than very large ones, thus the 51-100 and over 100 categories were combined.

The above sampling strategy prioritizes active contributors in order to survey respondents with potentially more recent interactions with projects, as well as to increase the likelihood of responses by surveying active GitHub users. However, there is an inherent trade-off in this sampling strategy, in that it may be potentially less likely to capture participants who have departed a project, such as due to conflict, an important variable in this study. To partially address this, developer activity is defined within a time frame of one year from data collection to aid in sampling participants who may have departed due to recent issues. However, because the survey is a self-report measure, the survey questions are designed with reference to the project respondents are currently most active in. Developers may contribute to one or more projects simultaneously, and this tool will not be able to capture situations in which developers are active in one project, but have left another project due to disagreements and therefore are no longer active in it to maximize participant recall. The results section will present analyses that attempt to highlight the extent to which the collected data set is sensitive to this potential sampling issue, and the implications of this will be further discussed in study limitations.

Procedure

Sampled participants were invited to participate by e-mail using the SoSciSurvey.de survey tool [2] at the end of April 2015. The invitations contained information about the purpose of the study, affiliation and contact information, together

[2] SoSciSurvey.de is an online hosted survey tool similar to Qualtrics and SurveyMonkey. It is developed and maintained by Dr. Dominik Leiner and available for free unlimited use for non-commercial purposes. This tool was chosen for its robust features, such as allowing the present researcher to customize the survey text based on responses given by participants and the ability to manage survey invitations, pilot and final survey data in one location.

with a personalized survey link. One reminder e-mail was sent a week later to respondents who have not yet clicked on their link. In exchange for participation, at the end of the survey, respondents could nominate a FOSS project to receive a $100 donation and informed that three projects will be chosen at the end of the study. All pilot participants received this option as well. Participants were also given the option to stay informed of the study results. As participants may be members of several projects at the same time, they were instructed to answer the questionnaire from the perspective of the project they contribute most actively to. This instruction was repeated across all questionnaire pages.

Demographics

A total of 520 responses were collected over 3 weeks. Of these, 228 were complete, passed attention checks and were used for further analysis. Responses were received from 59 countries, with USA the largest group (33%) followed by Germany (8%), and the UK (6%). Consistent with earlier work, the sample was largely male – only 7% identified as "Female", and 3% identified their gender in another way, including "Gender Queer", "Feminine Male" and "Jedi". The sample was balanced in terms of developer tenure and roles within the project: 28% have contributed to their project for over 4 years, 31% for less than 1 year and the remaining between 1 and 4 years. 32% were project leaders or founders, 34% core contributors or maintainers and 21% were members with commit access. 46% of projects were small (between 2 and 10 developers), 17% were very large (more than 50 contributors) with the remaining in between.

5.2 Results

5.2.1 Model Specification: Measurement Model

A second exploratory factor analysis (EFA) employing Varimax rotation was performed on only the cleaned survey responses (n=228). The result suggested an expected 4-factor structure that was a significant improvement over the pilot study (Table 5.3). It explained 64% of the variance.

To verify the validity of the measurement model, a confirmatory factor analysis (CFA) was conducted next. All the four conflict variables were included as specified in the EFA (Table 5.3) together with the antecedent and outcome variable measures. The lavaan package in R (version 0.5-17) and full information maximum likelihood (FIML) estimation were used to run the CFA and subsequent structural equation modeling (SEM) analyses on a covariance matrix. FIML was used over listwise deletion because it has been found to perform better across all forms of missing data (R. P. McDonald & Ho, 2002). Some indicators had low factor loadings in the initial analysis, thus all indicators with loadings less than .55 were removed. Standardized factor loadings in the final model ranged from .62 to .95. Table 5.4 shows the factor loadings in the final measurement model used.

The final measurement model had a good fit as Table 5.5 demonstrates. Though the Chi-square p value is typically expected to be above >.05, this test is sensitive to sample size and number of variables (R. P. McDonald & Ho, 2002). Thus the remaining size independent fit indices are considered instead, which are all within acceptable ranges (CFI and TLI > .90, RMSEA < .05, SRMR <.08) (R. P. McDonald & Ho, 2002). The four-factor measurement model was also compared against a model with all conflict indicators loading on one factor, and the four-factor solution was found to have a significantly better model fit.

Table 5.3: Exploratory Factor Analysis using Study Data

Items	NORM	TASK	AFFECT	PROC
There is friction among contributors	.19	.29	**.78**	.20
Personality conflicts are evident	.15	.22	**.82**	.18
There is interpersonal tension among contributors	.25	.16	**.81**	.28
There are emotional displays (i.e. swearing, flaming, capitalization) among contributors	.30	.00	**.62**	.06
Contributors have different opinions regarding the work being done	.06	**.78**	.22	.04
There are differences of opinion in this project	.15	**.83**	.11	.02
There is conflict among contributors about the work being done	.35	**.50**	.45	.24
There are there conflicts about ideas in this project	.41	**.63**	.17	.11
There are differences of opinion about what features should be added	.23	**.75**	-.04	.20
Contributors disagree on proposed changes to this project's code base	.41	**.54**	.23	.10
Contributors disagree about coding style	**.59**	.19	.20	.06
There are disagreements about who should do what in this project	.12	.12	.15	**.89**
There is conflict among contributors about task responsibilities	.20	.13	.15	**.87**
Contributors disagree about resource allocation	.27	.12	.19	**.69**
There are disagreements about authority in this project	.35	.01	.35	**.60**
Contributors to this project disagree about the project mission	**.56**	.34	.11	.29
There are disagreements concerning the values this project upholds	**.76**	.18	.21	.08
Contributors disagree about what is expected behavior in this project	**.74**	.15	.27	.14
Contributors disagree about the tone and style of communication used within the group	**.61**	.01	.46	.23
There are ideological disagreements	**.62**	.20	.18	.23
Contributors disagree about conventions used in this project	**.71**	.20	.18	.23
Contributors debate the agreed upon customs and norms of this project.	**.60**	.20	.03	.27

Extraction Method: Principal Component Analysis. Rotation Method: Varimax with Kaiser Normalization.
a. Rotation converged in 6 iterations

Table 5.4: Results of Confirmatory Factor Analysis (CFA)

Items	Stand. Loadings	Items	Stand. Loadings
AFFECT1	.81	IDENT1	.68
AFFECT2	.85	IDENT2	.67
AFFECT3	.91	IDENT3	.82
TASK1	.75	IDENT4	.68
TASK2	.81	INTER1	.78
TASK3	.64	INTER2	.80
PROC1	.90	INTER3	.80
PROC2	.93	INTER4	.63
PROC3	.64	DISTRIB1	.76
NORM1	.76	DISTRIB2	.85
NORM2	.83	DISTRIB3	.70
NORM3	.76	LEADER1	.80
PERF1	.81	LEADER2	.83
PERF2	.73	LEADER3	.79
PERF3	.68	LEADER4	.80
ITR1	.76	LEADER5	.83
ITR2	.83	DECIS1	.67
ITR3	.63	DECIS2	.94
		DECIS3	.80

Table 5.5: Fit indices for measurement model with four conflict factors, a one factor solution and structural model

Model (N=222)	Fit Indices				
	χ^2/df	CFI	TLI	RMSEA	SRMR
4-Factor CFA	1.45	.93	.92	.05	.06
1-Factor CFA	2.35	.79	.77	.08	.08
Structural model	1.5	.92	.91	.05	.06

Finally, the convergent and discriminant validity of all factors in the measurement model were tested by examining the Average Variance Extracted (AVE) and Square Maximum Correlations (SMC). Table 5.6 presents the AVE scores for the model variables, with all values above .50 showing that the indicators accounted for more variance in the latent factors than standard error, confirming their convergent validity. Additionally, Table 5.6 shows that AVE was larger than the squared maximum correlations (SMC) between the variables, confirming discriminant validity of the factors. Table 5.6 also reports the composite reliability[3] (CR) for each latent factor.

[3]Composite reliability is presented rather than the traditional Chronbach's alpha because this statistic is more commonly used in Structural Equation Modelling approaches, and because alpha is known to underestimate construct reliability (Peterson & Kim, 2013).

Table 5.6: Means, Standard Deviations, Reliability and Validity Statistics for Final Survey

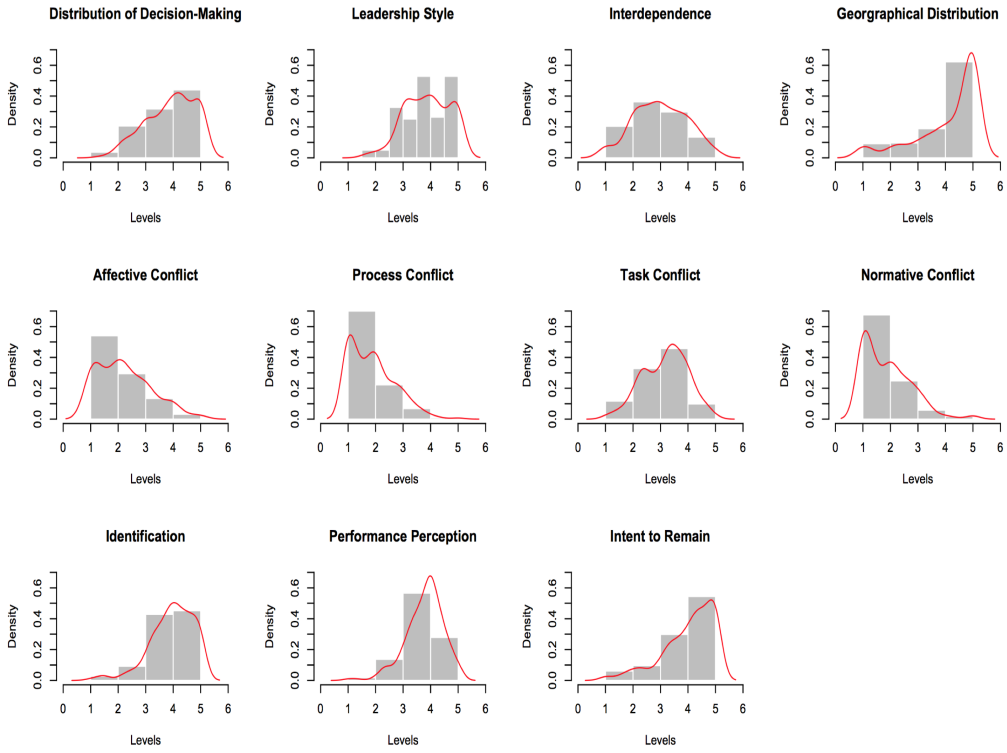
Factor	Mean	Standard Dev	CR	AVE	SMC
AFFECT	2.22	1.00	.90	.75	.32
TASK	3.06	0.82	.81	.60	.30
PROC	1.88	0.82	.87	.70	.23
NORM	1.88	0.87	.84	.63	.32
PERF	3.78	0.68	.79	.56	.16
ITR	4.02	0.95	.79	.56	.21
IDENT	3.96	0.80	.81	.52	.21
INTER	3.01	0.96	.84	.57	.06
DISTRIB	4.10	1.17	.80	.57	.11
LEADER	3.87	0.82	.91	.67	.19
DECIS	3.88	0.89	.84	.65	.21

These findings address the first research question and show that normative conflict emerges as a distinct construct separate from task, affective and process conflict. The findings also show that all other measures in the measurement model were of good quality and able to discriminate between the different constructs tested.

5.2.2 Descriptive statistics

Table 5.6 also presents the means and standard deviation for the latent variables used in the final model. In general, overall conflict levels reported are fairly low: task conflict occurs most frequently (M=3.06, SD=0.82) while affective (M=2.22, SD=1.00), procedural (M=1.88, SD=0.82) and normative conflict (M=1.88, SD=0.87) less so. On the other hand, identification with the team (M=3.96, SD=0.80), performance perception (M=3.78, SD=0.68) and intent to remain (M=4.02, SD=0.95) in the project are relatively high overall. Teams surveyed are also highly distributed geographically (M=4.10, SD=1.17), although this variable had a wide spread, and have largely participative decision making styles (M=3.88, SD=0.89) and transformational leaders (M=3.87, SD=0.82). Full distributions are available in Figure 5.1.

Figure 5.1: Distribution of responses across key variables



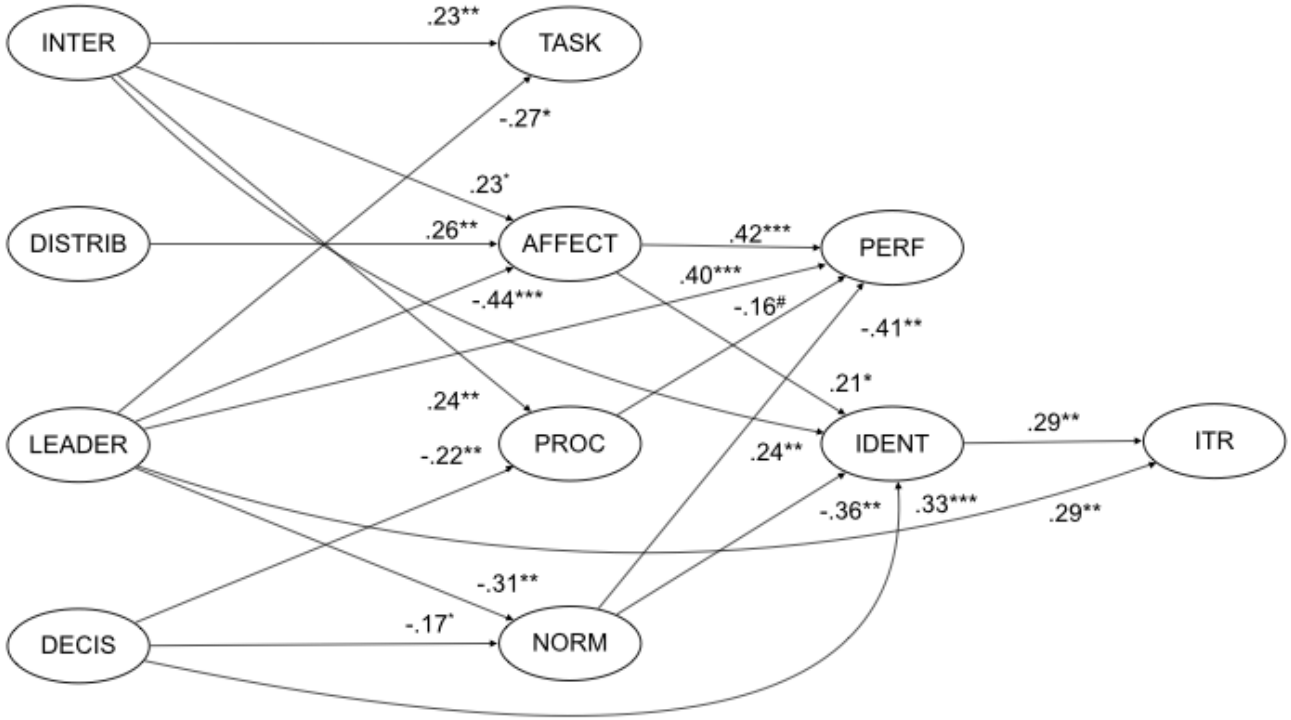
5.2.3 Model Specification: Structural Model

After confirming the validity of the measurement model, a structural equation model (SEM) was used to investigate the hypothesized relationships indicated in Figure 5.2. Table 5.5, presented earlier, shows that the structural model had good fit. Figure 5.2 summarizes the significant hypothesized relationships uncovered by the analysis.

5.2.4 Hypothesis testing

Table 5.7 shows a summary of the various hypotheses tested, and whether they were supported. The sections below present the results, starting with the relationship between conflict, emergent states and outcomes. Following this, the relationship between conflict dimensions and their antecedents is examined. Interactions between conflict types and their impact on emergent states and outcomes are also presented. Finally more complex mediation effects involving antecedents, conflict and emergent states are explored.

Figure 5.2: Structural Equation Model using Survey Data



* p < 05, ** p < .01, *** p < .001

Table 5.7: Hypotheses and findings

Hypothesis	Result
H1: Developers' identification (IDENT) will be positively related with intention to remain in the project.	**Supported**
H2a: Task conflict (TASK) will be positively related with performance perception in FOSS development teams.	**Not supported**
H2b: Moderate levels of task conflict (TASK) will be more positively related with performance perception in FOSS development teams than high or low task conflict levels.	**Not Supported**
H3a: Affective conflict (AFFECT) will be negatively related with perception of project performance	**Not Supported**
H3b: Affective conflict (AFFECT) will be negatively related with team identification.	**Not Supported**
H4: Process conflict (PROC) will be negatively related with perception of project performance.	**Supported**
H5a: Normative conflict (NORM) will be negatively related with performance perception	**Supported**
H5b: Normative conflict (NORM) will be negatively related with team identification	**Supported**
H5c: Identification will mediate the relationship between affective conflict and intention to remain.	**Not Supported**
H5d: Identification will mediate the relationship between normative conflict and intention to remain.	**Supported**
H6a: Interdependence (INTER) will be positively related with the occurrence of task, affective, and process conflict.	**Supported**
H6b: Conflict will mediate the positive relationship of interdependence and identification with the team	**Not Supported**
H7: Geographically distributed (DISTRIB) teams will experience more task, procedural and affective conflict	**Partially Supported**
H8a: Transformational leadership (LEADER) will be negatively associated with all types of conflict	**Partially Supported**
H8b: Conflict will mediate the positive relationship between transformational leadership (LEADER) and team identification (IDENT)	**Supported**
H9a: Participatory decision-making (DECIS) will be negatively related with affective, process and normative conflict	**Partially Supported**
H9b: Conflict will mediate the positive relationship between participatory decision-making (DECIS) and team identification	**Not Supported**

Conflict and performance perception

H2a proposed that task conflict would have a positive relationship with perceptions of team performance. However, results of the structural equation model show that task conflict does not have a significant relationship with performance perception (β=-.03, p>.05). Therefore H2a is not supported.

H3a predicted that affective conflict would have a negative relationship with performance perception. However, contrary to predictions, affective conflict was found to have a large[4] and significant positive effect on perception of team performance (β=.42, p<.01). In other words, greater levels of affective conflict were related with more positive perceptions of team performance. Therefore H3a is not supported.

Furthermore, H4 predicted that process conflict would have a negative relationship with performance perception. Results show a marginally significant, moderate and negative effect of process conflict on performance perception, (β=-.16, p=.07), in partial support of H4.

Finally, H5a predicted that normative conflict would have a negative relationship with perceptions of team performance. Findings show that the relationship between normative conflict and performance perception is strong, significant and negative (β=-.41, p<.01), thus H5a is supported. In other words, the presence of greater normative conflict levels reduces the positive perception of the team's performance.

Taken together, the findings show that normative conflict and process conflict have a negative impact on developer's perceptions of team performance, while affective conflict appears to be positively connected with performance perception.

[4]Effect sizes are interpreted based on guidelines by Cohen (1992) on multiple correlations and multiple partial correlations, as recommended by Durlak (2009).

Conflict, identification and intention to remain

H3b proposed that affective conflict would be negatively connected with team identification. Results of the structural equation model show that, contrary to predictions in H3b, affective conflict has a significant, moderate and positive relationship with identification with the team (β=.21, p<.01). Therefore, H3b is not supported. Furthermore, H5b proposed that normative conflict would be negatively connected with team identification. Results show that normative conflict is strongly and negatively related to identification with the team (β = -.36, p <.01), supporting H3b.

Additionally, H1 proposed that identification with the team would be positively related with intention to remain in the project. Results of the structural equation model show that identification is significantly, moderately and positively related to intention to remain in the project (β = .29, p <.01), supporting H1. Taken together, affective conflict and normative conflict appear to indirectly and differentially impact intention to remain by acting on identification (H5c and H5d).

To test this potential indirect relationship, the present study uses the delta method together with the existing structural equation model. Baron and Kenny (1986) recommend four essential steps to establish mediation: 1) establishing a significant relationship between the independent variable and outcome, 2) a significant relationship between the mediator and outcome, 3) establishing that the mediator has an effect on the outcome in the presence of the interdependent variable, and 4) for full mediation, the effect of the independent variable on the outcome variable would reduce to zero in the presence of the mediator. These steps are recommended when using a series of hierarchical regressions. However, structural equation models have the advantage of simultaneously specifying relationships between several levels of variables, thus simplifying mediation testing significantly. Using R and lavaan, it is only necessary to include an additional Sobel test (also known as the delta method) (Sobel, 1982) in the model

specification, for the model to estimate both direct and indirect effects, and their total impact on the outcome variable. The Sobel test is more conservative than the bootstrapping technique, which is also commonly used. The test does not affect model fit or the values of model coefficients, thus it was conducted together with the structural model.

H5c predicted that identification would mediate the relationship between affective conflict and intention to remain. Results of the Sobel test show that affective conflict does not have a significant direct effect on intention to remain (β=.07, p>.05), while the indirect effect through identification is also not significant (β=.06, p>.05). Thus no mediation is found, and H5c is not supported.

H5d predicted that identification would mediate the relationship between normative conflict and intention to remain. Results show that normative conflict has a significant, small and indirect relationship with intention to remain through identification (β=-.10, p<.05), while the direct relationship between normative conflict and intention to remain is not significant (β=-.19, p>.05). Therefore, identification fully mediates the relationship between normative conflict and intention to remain, supporting H5d. In other words, normative conflict has an effect on intention to remain only indirectly through reducing identification levels.

Taken together, these findings show that both affective conflict and normative conflict have significant and differential impact on identification. Affective conflict is positively related with greater team identification, while normative conflict reduces developer identification with the team. Furthermore, though normative conflict does not have a direct relationship with intention to remain, it indirectly affects intention to remain by reducing identification levels with the team. Normative conflict therefore emerges as the only conflict dimension to predict intention to remain in the project.

Conflict Antecedents

H6a predicted that team interdependence will be significantly and positively related with the occurrence of task, affective and process conflict. Results show that team interdependence does indeed have a positive, moderate and significant relationship with task (β=.23, p<.01), affective (β=.23, p<.05) and process (β=.24, p<.01) conflict in FOSS teams, thus supporting H6a. In other words, the more interdependent FOSS teams are, the more likely they are to experience task, affective and procedural issues.

H7 predicted that more geographically distributed teams would experience greater task, affective and process conflict. Results show that only affective conflict has a significant, moderate and positive relationship with geographical distribution (β=.26, p<.01). Task conflict has a marginally significant and moderate positive relationship with geographical distribution (β=.19, p=0.06), while normative conflict shows no significant relationship (β=.05, p=0.63). Thus H7 is only partially supported.

H8a proposed that transformational leadership would reduce the occurrence of all four conflict types. Results show that leadership style is significantly and negatively related to the occurrence of task (β=-.27, p<.05), affective (β=-.44, p<.001) and normative (β=-.31, p<.01) conflict only, in partial support of H8a. Therefore, a greater tendency toward a transformational leadership style moderately reduces the occurrence of task, and strongly reduces the occurrence of affective and normative conflict.

Finally, H9a proposed that more distributed decision-making in the team would reduce the occurrence of affective, process and normative conflict. Results show that distribution of decision-making is significantly, moderately and negatively connected with the occurrence of process (β=-.22, p<.01) and normative (β=-.17, p<.01) conflict only, in partial support of H9a.

Interaction Effects

As the four types of conflict may have potential interactions with one another, additional tests were also performed to investigate moderation effects, thereby addressing the research question (RQ2). Though it is possible to include interaction terms in an SEM model, this procedure makes effects interpretation more difficult. Additionally, there are 6 potential interactions possible for each outcome, and including these in the model would make the research model overly complex. Thus the analysis was performed post-hoc using hierarchical multiple regressions instead. This also allowed the inclusion of control variables: gender and role in the project were coded as dummy variables, while length of contribution (in years) and team size were treated as continuous, and included in the first step. In the second step, the four conflict variables were entered. In the third step, all 6 possible interaction effects between the 4 conflict types were entered.

Residual centering was used to compute the interactions, rather than traditional mean centering, to allow for easier interpretation of the main and interaction effects. Residual centering allows the computation of interaction terms that are orthogonal to their respective main effects (i.e. have no effect), and thus allow main effects to be interpreted together with their interaction effects (Little, Bovaird, & Widaman, 2006). To compute the interaction terms, procedure proposed by Little et al. (2006) was used: the product of two unscented conflict composite variables (e.g. Task*Affective) was regressed onto the two main conflict variables. Then residuals from this regression were saved as a new orthogonal interaction variable (Task*AffectiveR).

Regression results (Table 5.8) show that among the four conflict types, only normative conflict has a direct and negative main effect (β= -.25, p <.01) on identification with the team. Unlike the SEM model, affective conflict does not have a significant main effect on identification in the regression model, possibly because the regression does not control for the direct effects of antecedents

on outcomes the way the SEM model does. Among the control variables, only length of contribution (β = .19, p<.01) and the activity level per week (β = .17, p<.05) have a significant positive relationship with identification, while committers (β = -.26, p<.01) and members with no official title (β =-.18, p<.05) report significantly less identification compared to team leaders. There are no significant interactions between conflict types associated with identification.

Table 5.8: Regression predicting identification and performance perception based on conflict dimensions and their interactions

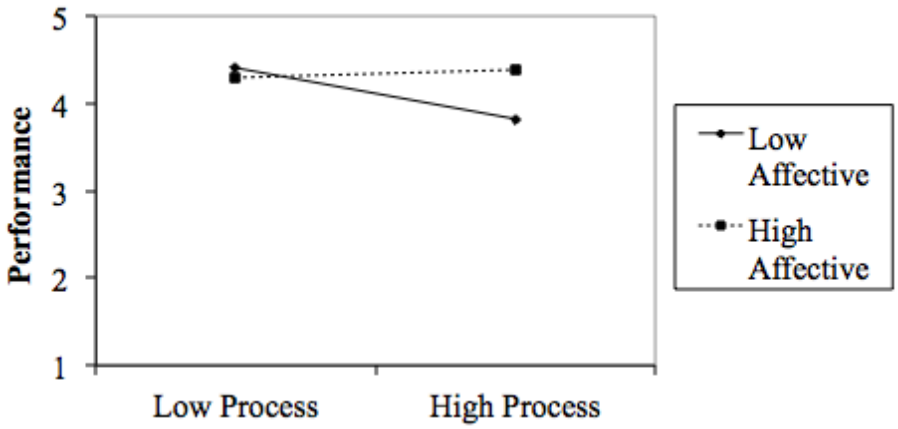
Block	Predictors	Standardized Beta					
		Perf.	Perception		Identification		
1	Gender: Male +	.07	.05	.05	.09	.08	.05
	Gender: Other	.04	.01	.01	.01	.00	-.01
	Role: Core Team #	-.13	-.11	-.11	-.11	-.10	-.10
	Role: Committer	-.04	-.09	.09	-.23**	-.26**	-.26**
	Role: No Official Title	-.02	-.01	-.01	-.19*	-.17*	-.18*
	Years Contributed	.12	.10	.10	.22**	.19**	.19**
	Activity Level	.03	.00	.00	.19**	.18*	.17*
	Number of Developers	.10	.13	.14	-.13ˆ	-.11	-.12
2	Affective Conflict		.19*	.20*		.09	.08
	Task Conflict		-.07	-.11		.04	.05
	Process Conflict		-.15ˆ	-.15ˆ		-.09	-.08
	Normative Conflict		-.27**	-.26**		-.27**	-.25**
3	Process * Affective R			.22**			.04
	Task * Affective R			-.07			-.11
	Task * Process R			-.10			.07
	Task * Normative R			.26**			.07
	Process * Normative R			-.05			-.07
	Normative * Affective R			-.13			.09
	Model R2	.04	.15	.22	.18	.25	.26
	F Change	1.56	6.19**	2.55*	2.91ˆ	4.28**	0.55

ˆ p<.10, * p<.05, ** p<.01
+ Relative to excluded Gender: Female; # Relative to excluded Role: Leader

Affective (β = .19, p<.05), process (β = -.16, p<.05) and normative (β = -.25, p<.01) conflict have significant main effects on performance perception. None of the control variables have significant main effects on performance. There is also a significant and positive interaction between affective and procedural conflict (β = .22, p<.05). Figure 5.3 details this two-way interaction – at low levels of affective conflict, greater procedural conflict has a negative effect on performance but at high levels there is no difference. Therefore, groups that

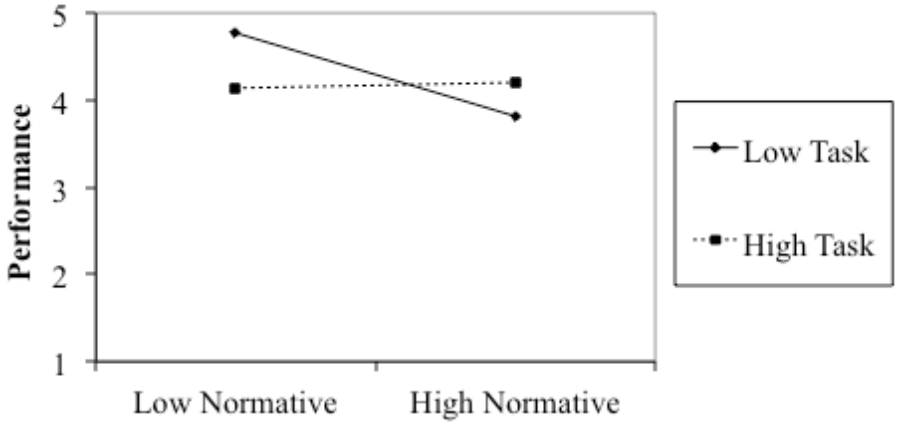
experience both process and affective conflict are perceived to perform better than groups that experience predominantly process conflict.

Figure 5.3: Interacting effects of affective and process conflict on performance perception in FOSS teams



There is also a significant and positive interaction between normative and task conflict (β = .224, p<.05). As shown in Figure 5.4, at low levels of task conflict, normative conflict has a more negative effect on performance perception, than at high levels of task conflict. Therefore, teams that experience normative conflict together with task conflict are perceived to perform better than teams experiencing only normative conflict.

Figure 5.4: Interacting effects of normative conflict and task conflict on performance perception of FOSS teams



Conflict as mediator between antecedents and emergent states

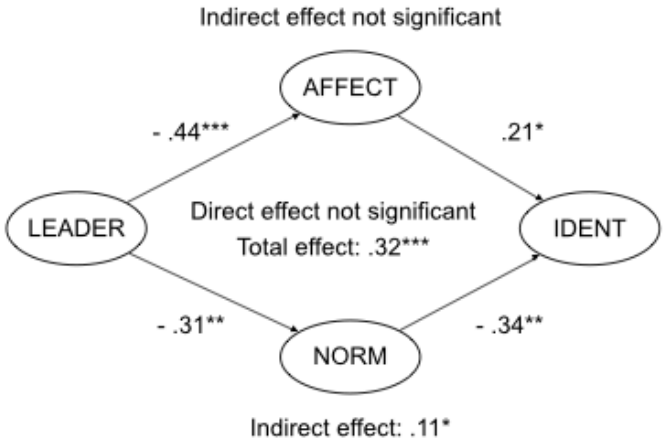
As there appear to be both direct and indirect effects of antecedent variables on outcome variables, the study also investigated possible mediation effects between antecedents, conflict and emergent states. A similar procedure was used to the mediation effects between conflict, identification and intention to remain as described above. That is, the Sobel test was included together with the structural equation model (Sobel, 1982).

H6b proposed that conflict would mediate the positive relationship of interdependence and identification with the team. Only affective conflict was significantly connected with both interdependence (β=.24, p<.05) and identification (β=.21, p<.01) in the structural model, thus this is the only mediation effect tested in connection with this hypothesis. Results of the mediation analysis show that interdependence has a significant direct relationship with identification (β=.24, p<.01), however the indirect relationship between interdependence, affective conflict and identification is not significant. Thus no mediation is found, and H6b is not supported.

H8b proposed that conflict would mediate the positive relationship between transformational leadership and team identification. Both affective and normative conflict were significantly connected with leadership (β=-.44, p<.001; β=-.31, p<.01) and identification (β=.21, p<.05;β=-.36, p<.01;) in the structural model, thus both mediators were included simultaneously in the mediation equation. Figure 5.5 is provided below to illustrate this more complex mediation relationship.

Mediation results show that both the direct effect of leadership on identification, and the indirect effect of leadership through affective conflict are not significant. However, the indirect effect of leadership on identification through normative conflict is significant. Therefore normative conflict fully mediates the relationship between leadership and identification, supporting H8b. In other words, transformational leadership increases identification with the team (and

Figure 5.5: Mediation effects between transformational leadership, affective conflict, normative conflict and identification with the team



* p < 05, ** p < .01, *** p < .001

therefore, intention to remain in the project), indirectly by reducing normative conflict levels.

Finally, H9b proposed that conflict would mediate the positive relationship between participatory decision-making and team identification. Only normative conflict had a significant relationship with both distributed-decision making (β=-.17, p<.05) and identification(β=.36, p<.01), thus it is the only mediation analysis conducted for this hypothesis. Mediation results show that distributed decision-making has a significant direct effect on identification with the team (β=.33, p<.01), however, no significant indirect effect is found involving normative conflict. Thus no mediation is found, and H9b is not supported.

Taken together the findings show that team structural variables such as leadership style, task interdependence, and distribution of decision-making style have varying relationships with team outcomes in the presence of affective and normative conflict. Specifically, the extent to which a team leader employs the transformational leadership style has an indirect effect on identification, in the presence of normative conflict. Given that the structural equation model has shown that identification is the only predictor of intention to remain, these findings suggest that when normative conflict is present, it is essential to address it in order to prevent developer turnover. Importantly, factors such as task in-

terdependence and distribution of decision-making continue to positively and directly affect identification with the team, and consequently developer intention to remain in the project, even in the presence of conflict.

Quadratic Effects

The results of both SEM and regression analyses showed that task conflict did not have any significant relationship with outcome variables. The lack of significant effects is surprising given that prior research shows task conflict to have a positive relationship with team outcomes in cognitively diverse settings (Arazy et al., 2011). Research has also shown that task conflict can have a curvilinear relationship with performance (De Dreu, 2006). Specifically, while low and high levels of task conflict can have a negative effect on performance, moderate levels are beneficial for team performance. Thus H2b proposed a quadratic relationship between task conflict and performance perception, such that moderate levels of task conflict would be more positively related to performance perception than high or low levels.

To test for a possible quadratic effect, the square of the original task conflict variable was entered into the third step of the same hierarchical equations predicting performance and identification (Table 5.9). The squared term was not significant in either regression, and it did not result in significant changes to the main effects and interactions reported above. Thus, in contrast to earlier work (De Dreu, 2006; Tekleab et al., 2009), results did not show a curvilinear relationship between task conflict and performance perception. Thus H2b was not supported.

Table 5.9: Regression analyses predicting performance perception and identification including quadratic effects

Block	Predictors	Standardized Beta					
		Perf. Perception			Identification		
1	Gender: Male +	.05	.06	.05	.08	.08	.05
	Gender: Other	.01	.02	.02	.00	.00	-.01
	Role: Core Team #	-.11	-.09	-.11	-.1	-.10	-.10
	Role: Committer	-.09	-.07	-.09	-.26**	-.26**	-.26**
	Role: No Official Title	-.01	-.01	.00	-.17*	-.17*	-.18*
	Years Contributed	.10	.06	.10	.19**	.19**	.19**
	Activity Level	.00	.00	.00	.18*	.17*	.17*
	Number of Developers	.13	.13	.14ˆ	-.11	-.11	-.12
	Affective Conflict	.19*	.19*	.20*	.09	.09	.08
	Task Conflict	-.07	-.76ˆ	-.36	.04	-.11	.04
	Process Conflict	-.15ˆ	-.15ˆ	-.15ˆ	-.09	-.08	-.08
	Normative Conflict	-.27**	-.30**	-.27**	-.27**	-.27**	-.25**
2	Task Conflict 2		.72	.26		.15	.00
3	Process * Affective R			.22**			.04
	Task * Affective R			-.09			-.11
	Task * Process R			-.11			.07
	Task * Normative R			.24*			.07
	Process * Normative R			-.04			-.07
	Normative * Affective R			-.12			.09
	Model R2	.15	.16	.22	.25	.25	.26
	F Change	2.91**	2.89ˆ	2.07ˆ	5.40**	0.15	0.01

ˆ p<.10, * p<.05, ** p<.01
+ Relative to excluded Gender: Female; # Relative to excluded Role: Leader

5.3 Discussion

The present study examined the sources and effects of different types of conflict in Free and Open Source (FOSS) development. Specifically, the study aimed to 1) quantitatively examine the relevance of normative conflict to the intragroup conflict taxonomy and validate the updated four-factor conflict structure, 2) use the updated framework to investigate how different conflict types influence sustained developer participation through their perceptions of team performance and identification and 3) explore the structural antecedents to different conflict dimensions experienced in FOSS projects. The study found that normative conflict emerged as a distinct fourth dimension alongside task, process and affective conflict. While both process and normative conflict were negatively connected with performance perception, normative conflict emerged as the only conflict dimension to be negatively connected with intention to remain in the project by acting indirectly on identification. In general, interdependence and geographical distribution were associated with greater levels conflict, while transformational leadership and distributed decision-making were associated with lower conflict levels. Details of these findings, as well as their implications for theory and practice are discussed below.

5.3.1 Normative Conflict as a Separate Conflict Dimension

Using several pre-tests and a wide survey, the study has developed and validated a normative conflict measurement scale, and reformulated the intragroup conflict taxonomy to include this new dimension. Findings show that normative conflict stands out as a distinct conflict dimension, supporting results in Study 1 and addressing the first research aim.

While there already exists a validated multi-dimensional scale for affective, task and process conflict from Jehn and colleagues (1999), the present study shows it is helpful to adapt existing measures to the context being studied. Sim-

ilarly, while there also exist separate measures for normative conflict, they are either focused on internal perceptions of normative dissonance within an individual (R. I. McDonald et al., 2012) or the amount of times an individual disagreed with the group (Packer & Chasteen, 2010) rather than measuring the explicit overall manifestations of normative conflict within the team. As Chapter 3 has shown, normative conflict may involve not only a single persons' propensity to disagree with the majority, but can also manifest as an active discussion involving several members concerning how well the overall group behavior (descriptive norms) matches the group goals (injunctive norms). Thus the normative conflict scale is useful in measuring more varied manifest examples of conflict. This is also the first study to combine the four conflict types into one validated and parsimonious scale.

The resulting expanded conflict scale affords researchers into the FOSS phenomenon a consistent measurement tool relevant to the study context. The updated taxonomy can be used to support and contrast findings from future research into online collaboration that examines additional important input and output variables. Furthermore, because the updated conflict taxonomy is an expansion of an existing scale validated and widely used in organizational studies, it also affords contrasting findings across ongoing teams in different organizational settings. Chapter 6 will examine these differences with reference to findings from both Study 1 and Study 2.

5.3.2 Differential impact of conflict types on team emergent states and outcomes

In addressing the second study aim, the work explicates the impact of the updated conflict taxonomy on sustained participation and team emergent states such as team identification and performance perception. A number of interesting findings emerge.

Intention to Remain

First, the present study shows that normative conflict is the only dimension to negatively impact sustained participation, when controlling for individual and team differences. Specifically, normative conflict indirectly reduces developer intention to remain in the project by decreasing identification with the team. This finding is interesting relative to previous literature on normative conflict. As Packer (2007; 2010) shows, individuals are more likely to express normative dissent against a group when they identify more strongly with the group in question. Thus the stronger the sense of belonging, the more likely an individual is to speak up for the benefit of the group when they perceive a normative dissonance to occur. The present study shows that the effect of this sort of dissent on other group members is the reverse: greater levels of manifested normative conflict within a group stimulate a reduced identification with the group.

Because normative conflict highlights a dissonance between community best practices and actual team behavior, it may reduce developers' sense of group cohesion, affecting their identification with the group. Furthermore, in line with Packer's (2007) prediction, members who have lower identification levels are more likely to disengage and leave the group. Consequently, developers whose sense of belonging is reduced due to the presence of normative conflict in the group have a lower intention to continue contributing to the project. It is possible that by highlighting normative dissonance between what the group intends to do and actual behavior, normative conflict may also help to crystallize subgroups of opinions within the community, either in support of, or against a particular activity. This may eventually lead to a project fork if enough members converge on a particular subgroup. This finding therefore reinforces the qualitative data in Chapter 3, and highlights the role of normative conflict in influencing team structure, such as the formation of faultlines through reduced group identification. It would be interesting for future work to elucidate this process in greater detail, such as by drawing on faultline research (Lau & Murnighan, 2005) to

measure changes in team level and subgroup identification following manifest normative dissonance.

Second, the study finds that affective conflict has a surprisingly positive relationship with intention to remain through increasing team identification. Thus, paradoxically, developers who perceive greater levels of affective conflict also identify most strongly with the group and intend to continue contributing. This finding stands in contrast to earlier work on affective conflict across different organizational types that predicted largely negative effects on team outcomes and member satisfaction (De Dreu & Weingart, 2003; de Wit et al., 2012). One reason for this difference could be the nature of participation in voluntary distributed teams as compared to virtual teams. Specifically, the presence of affective conflict in voluntary communities like FOSS may suggest greater emotional involvement in the project, thus indirectly signaling a successful project. Higher affective conflict levels may thus also be indirect indicators of project activity. As Dabbish and colleagues (2012) found, even potentially negative indicators such as visibility of membership turnover can lead to greater participation intentions because turnover demonstrates a certain level of activity within the group. The effect is particularly salient in groups with strong team identification who are better able to recover from affective disagreements due to their common sense of purpose. Thus, paradoxically, the present findings may be indicative that affective conflict is an indicator of an active and cohesive community.

Performance Perception

Third, the study contributes to literature on open collaboration by exploring the complex effects of different types of conflict on developers' perception of team performance. It is important to understand factors that influence developer perceptions of team performance, because it is an important indicator of overall project success (Campion et al., 1996). Specifically, the study finds normative and process conflict to be the strongest negative predictors of team performance

perception, while affective conflict shows a positive relationship.

This study proposed that normative conflict would be negatively connected with perceptions of project performance, because normative conflict has the potential to take attention away from the task at hand. Supporting this, the study finds a direct and negative relationship between normative conflict and performance perception. Furthermore, the study also finds an interaction between normative conflict and task conflict in predicting performance perception. To recap, at lower levels of task conflict, high levels of normative conflict have a greater negative impact on performance. Thus, interestingly, when there is a lack of debate about task related issues, such as adding new features or disagreeing about a proposed change to the code base, normative conflict has the potential to "steal the show" in community discussion and therefore lead to reduced performance. When taken together with earlier findings on reducing intention to remain in the project, normative conflict appears to have a largely negative, two-fold effect on team success. Thus normative conflict emerges as a valuable area for future research in understanding factors that detract from the success and longevity of ongoing virtual teams.

Findings also show that process conflict is negatively connected with performance perception, both directly as well as when interacting with affective conflict. Specifically, when more procedural conflict is present at lower levels of affective conflict in the team, developers perceive lower team performance. Thus process conflict distracts from team success both when present on its own or when other conflict types evolve a procedural dimension. This is consistent with work across different types of organizations, such as online collaborative systems like Wikipedia (Arazy et al., 2013) as well as findings on both ongoing virtual and ongoing collocated teams in traditional organizations (Jong et al., 2008; de Wit et al., 2012). Thus, taken together with previous work, the present findings highlight an interesting dynamic. Ongoing teams have a particular focus on efficiency and iterative improvement of team processes (Saunders

& Ahuja, 2006). At the same time, conflict about these procedural issues is distracting teams from the task at hand. Saunders and Ahuja (2006) suggest that ongoing teams have a greater focus on efficiency rather than performance, that is, doing things well rather than doing more things quickly, thus resulting in more process conflict. It may be illuminating to study this dynamic further in ongoing virtual teams as this may have practical implications like measuring performance in ongoing teams in a way that is consistent with the team's long-term focus.

Finally, the study did not find a significant effect of task conflict on performance, similar to recent prior work on Wikipedia (Arazy et al., 2013) and in contrast to other studies both on peer production and traditional virtual teams (Arazy et al., 2011; De Dreu & Weingart, 2003; de Wit et al., 2012). The possibility of a more complex curvilinear relationship was also investigated (Arazy et al., 2011; De Dreu, 2006), but findings did not show differing task conflict intensities to have a significantly different impact on emergent states. This may be because task conflict can become an expected part of the software development process. As Chapter 3 demonstrates, FOSS developers expect a certain level of critique when submitting patches. In fact, systems like GitHub and code review or "request for comment" practices are designed to stimulate feedback when contributions are submitted. Thus task conflict may be seen in FOSS projects as a part of the process, rather than an extraneous force on the team. It may be enlightening for future work to examine if the presence of a pro-feedback group norm would serve to moderate the relationship between task conflict and team or individual outcomes.

Taken together, these findings suggest that different kinds of conflict have differing and complex relationships with FOSS team outcomes, thus reaffirming the need to examine them individually. Additionally, while task conflict occurs most frequently, normative conflict has by far the most negative effects on FOSS team outcomes. As a result, different conflict episodes may require different

conflict management strategies, with particular attention toward normative and process conflicts rather than affective conflicts.

Conflict antecedents and mediating effects

Finally, the present study aimed to examine conflict antecedents that emerged in Chapter 3 and their complex relationship with both conflict emergence and team outcomes. This final aim contributes a practical dimension to the study by 1) highlighting specific structures that can promote and reduce conflict, thus informing team design, and 2) by specifying direct and indirect mechanisms through which these structures influence team emergent states and outcomes.

When examining group-level conflict antecedents, the present study finds that greater participatory decision-making significantly reduces levels of procedural and normative conflict in FOSS teams, and thereby helps to moderate their negative effects on individual performance perception and identification. These findings are consistent with work on team empowerment and shared leadership in short-term virtual teams and collocated teams that proposed a greater distribution of decision-making would be connected with more positive team outcomes (Conger & Kanungo, 1988; Hoch & Kozlowski, 2014; C. L. Pearce, 2006). Hence, teams with more hierarchical structures that experience high levels of disagreements about process or the team direction may consider delegating greater decision-making power to contributors.

Furthermore, results also show that a transformational leadership style is significantly related with lower levels of procedural and normative conflict in the team. This is also consistent with work across different organizational and virtuality settings that found transformational leadership to have a positive overall impact on the team (Avolio et al., 1999; Bass, 1990; Purvanova & Bono, 2009). Thus when selecting new team leadership, either by vote or through appointment by a former leader, FOSS teams should consider not just the individual(s) technical contribution but also their charisma and the extent to which they in-

spire other developers to follow them in taking the project further.

Leadership appears to have a complex relationship with outcomes in the presence of affective and normative conflict. On one hand, the effect of transformational leadership on identification (and consequently intention to remain in the project) is fully mediated by normative conflict. On the other hand, its effect on performance perception is partially mediated by normative conflict and continues to have a direct, positive and strong effect on performance perception even in the presence of conflict. Leadership in FOSS teams is therefore a complex activity in relation to intragroup conflict processes: during active normative debates, it only has a significant impact on intention to remain in the project when it is directed at managing normative conflict levels. This finding is helpful in informing leadership activities within FOSS teams, and suggests the need to dedicate more attention toward managing normative conflict when it is present. For instance, as Study 1 showed, leaders can serve as mediators bridging gaps in understanding between different sides in a conflict to prevent the emergence of faultlines. Leaders can also use their authority to step in and help to clarify group norms that are inconsistent or ambiguous, thus reducing the effect of normative conflict.

Interestingly, interdependence in ongoing virtual teams emerges as simultaneously positive and negative force. On one hand, interdependent teams experience greater team identification, greater task conflict, and therefore diversity of opinions, and affective conflict, which indirectly signals an active and cohesive group. These findings somewhat support earlier work in short-term virtual teams that predicted process conflict would increase affective conflict (Kankanhalli et al., 2007). On the other hand, greater interdependence increases process conflict levels that have a negative impact on performance perception in the team. These findings suggest that despite FOSS teams tendency toward modularization in individual tasks (Howison, 2009), some level of interdependence among contributors (such as sequential interdependence on maintainers to accept patches)

is helpful in creating more closely-knit communities with a greater diversity of opinions. However, communities that rely more on each other for outputs need to be aware of their greater potential toward process conflict, and proactively manage these episodes.

Furthermore, Langfred (2007) suggests that in the presence of conflict, more interdependent teams may restructure to promote more modularity as a means of avoiding further conflict. It would therefore be interesting to examine if interdependence can also act as an emergent state, because this would be especially relevant in self-organizing teams like FOSS teams. This would also affect causality and results interpretation, as interdependence may not always be strictly an input variable.

Finally, the present findings show that as predicted, greater geographical distribution increases affective conflict. This is somewhat consistent with prior work on short-term virtual teams, such as Rutkowski et al. (2007) who found higher levels of temporal disassociation (or dispersion across time zones) are more related to affective conflict. This is also consistent with predictions by Kankanhalli et al. (2007) based on observations of short-term virtual student teams that information overload may introduce affective conflict. However, contrary to both Rutkowski and colleagues' and Kankanhalli and colleagues' conceptualizations, affective conflict in this study is a positive force on the group, rather than a negative one. Thus in stimulating greater affective conflict, geographical distribution may be an indirect positive force on the team. This stands in contrast to arguments made by researchers like Cramton (2001) who expected greater geographical distribution and team heterogeneity to be linked with more coordination difficulties and misunderstandings. In fact, Study 1 suggests coordination difficulties with regard to time zone issues and communication overload do exist, but they do not appear to be quantitatively linked with more negative types of conflict and a reduction in positive team outcomes. Perhaps this is because coordination across different time zones and a great volume of commu-

nication are expected parts of the contribution experience for FOSS developers, thus, similar to task conflict, they may view this as a necessary part of the process. Alternatively, their ongoing nature and a sense of shared identity may be ameliorating these effects, as work by Hinds and Mortensen suggests (2005; 2001).

Taken together, findings show that conflict arises in FOSS teams from structural predispositions through a series of complex processes. Furthermore, certain structural variables continue to have an effect on emergent states even in the presence of conflict. In particular, greater interdependence is connected with greater team identification even in the presence of conflict. Similarly, greater transformational leadership has a direct effect on performance perception even when conflict is manifested. However, when a team experiences normative conflict, transformational leadership needs to be directed at reducing normative conflict effects in order to improve team identification. These mechanisms provide initial advice on important issues in the design and management of ongoing virtual teams: greater interdependence, distributed decision-making and transformational leaders all emerge as factors that support the team when it encounters conflict.

Addressing Sampling Limitations

As section 5.1.3 introduced, there is a potential inherent limitation in the sampling method employed in this study. Specifically, by focusing on active contributors' reports of projects they have contributed to most recently, the study risks under-sampling participants who have left projects due to conflict. This is particularly salient in light of the relatively low affective, process and normative conflict levels reported in this study. The present section evaluates and discusses the extent to which this issue may be present in the data set, and its implications.

Given that developers were defined as "active" in the present study if they have had a commit accepted within the past year, we would expect the sampling

strategy to account for developers who have joined the project less than one year ago, and who have had conflict experiences less than one year ago. Furthermore, given that developer tenure is fairly well distributed across the four categories employed (1 year or less, around 2 years, around 3 years, 4 years or more), the sampling strategy does account for developers who have been active for longer than the one year period defined by the study. However, the sampling strategy may under-sample developers with tenure of 2 or more years who have experienced conflict if they have left the project and therefore were not contacted for participation.

If this is indeed the case, we would expect to see a drop off in the volume of all four types of conflict reported after the one year mark. However, if conflict levels across tenure show consistency after the 1 year mark, that could indicate that the sample may better represent older developers' experience with conflict. Thus, in order to understand to what extent this limitation is present in the dataset gathered in this study, additional analyses were run to determine the distribution of conflict responses across different levels of developer experience.

Specifically, we examined the difference between average conflict levels reported by participants of different tenure. The mean values for each conflict type across different experience levels are presented in table 5.10. The table illustrates that conflict mean values do drop for developers with 2 or 3 years of tenure, however, they increase for participants with 4 or more years of experience to levels similar or higher than those reported by newcomers.

Table 5.10: Average conflict levels reported by conflict type and tenure of participants

Conflict Type	**1 year or less**	**About 2 years**	**About 3 years**	**About 4 years**
Task	3.15	2.98	3.11	3.39
Affective	2.30	2.02	2.00	2.45
Process	1.99	1.77	1.67	1.99
Normative	2.06	1.84	1.54	1.92

Despite this consistent pattern (illustrated in Figures 5.6-5.9), to understand if the trends observed were statistically significant, a series of one-way ANOVAs

compared conflict levels across four categories of tenure in the project. Results show that there was a significant effect of participant tenure on levels of task ($F(3,219) = 2.91$, $p < .05$) and affective ($F(3,220) = 2.85$, $p < .05$) conflict reported and a marginally significant effect on levels of normative ($F(3,218) = 2.58$, $p = .055$) conflict reported.

Post-hoc analyses using Tuckey's criterion [5] for significance confirmed that the average conflict levels reported by newcomers (1 year or less) and those with 4 years or more experience were not significantly different across all 4 types of conflict. Similarly, the drop in conflict levels reported by developers with around 2 years of experience in the project was not significantly different from levels reported by newcomers.

Interestingly, developers with 4 years of experience or more reported significantly more task (M=3.60) and affective (M=2.70) conflict than those with 2 years of experience (M = 3.18 and M=2.25, $p<.05$ and $p = .057$ respectively) suggesting that more experienced developers may perceive greater conflict levels. This may be because they accumulate more experience over time with different conflict episodes. Alternatively, developers may get tired of disagreements after participating in the same project for a long time, and thus perceive greater conflict levels than there may actually be present in interactions. This is an interesting area for future investigation.

[5] Tuckey's criterion is typically employed when all sets of pairwise comparisons need to be performed. Thus it was chosen for this post-hoc analysis, as we were interested in detecting any significant drop offs any time after the first year of tenure. In such a situation, the Bonferotti correction may prove too strict, potentially leading to a Type II error in failing to detect the drop off.

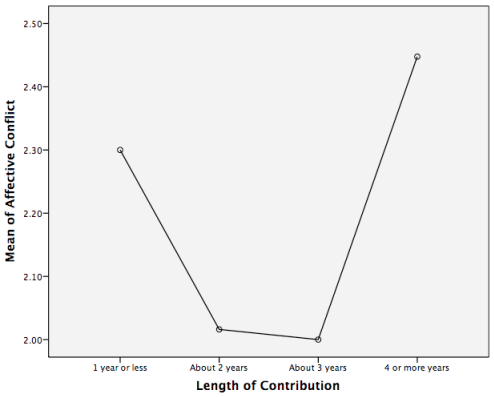


Figure 5.6: Affective conflict levels by developer tenure

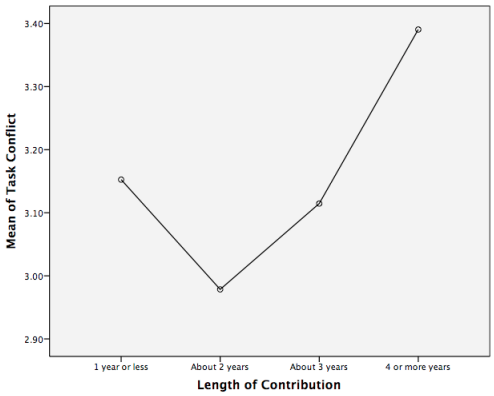


Figure 5.7: Task conflict levels by developer tenure

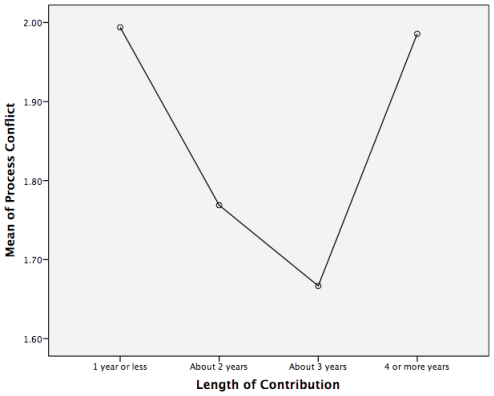


Figure 5.8: Process conflict levels by developer tenure

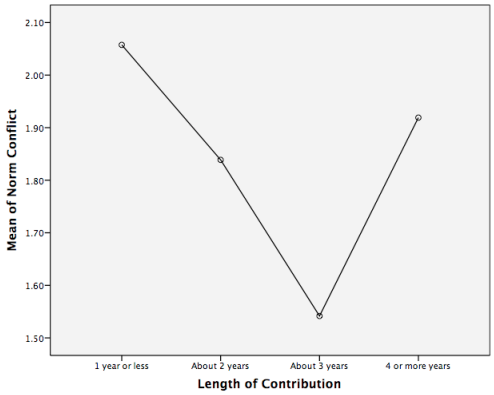


Figure 5.9: Normative conflict levels by developer tenure

The above results show that despite sampling limitations described above, the dataset is able to represent conflict levels for developers with various project tenure beyond the one year activity mark, and that developers with longer tenure and more conflict experience are less likely to be systematically left out.

To explore how these findings address the broader research aims of this thesis, the next chapter will bring together findings from both the present study and Study 1, discussing their relevance to the overall research aims as well as general implications for theory and practice.

Chapter 6

Discussion

6.1 Summary

This dissertation aimed to contribute to the study of conflict in three broad areas. First, the work set out to contribute to literature on the Free and Open Source Software phenomenon by being one of the first to systematically explore how conflict as a team processes is manifested, as well as its relationship with inputs, emergent states and team outcomes. In doing so, the work aimed to produce a consistent model of conflict that is both relevant to the FOSS context, and allows drawing parallels across studies on the phenomenon, as well as across different organizational settings. The work also aimed to provide practical recommendations for community development and management. Second, in situating this research in the FOSS context, the dissertation aimed to contribute to the growing body of work on virtual teams by exploring conflict specifically in ongoing distributed teams to understand how the process differs in conditions of expectations of future interaction and unbounded time frames. And third, the work aimed to illustrate the recursive mechanism through which conflict affects group development through the formation of norms and in serving as an input for future interactions.

To address these broad questions, two related studies were carried out. Study

1 featured a series of interviews with diverse representatives of the FOSS community, supplemented by participant observation and relevant public community interaction logs. Addressing the first aim, the study used a grounded approach to allow the emergence of conflict classifications that were native to FOSS contributors, as well as structural variables that may interact with different elements of the conflict process. This process has outlined four distinct conflict dimensions: task, process, affective and normative conflict; and their antecedents: interdependence, distribution of decision-making, leadership style, and geographical distribution. Addressing the second and third overall research aims, Study 1 further highlighted that conflict, in particular its normative dimension, is a recursive process that feeds back into team inputs in future interactions. Specifically, normative conflict allows FOSS teams to reflect on and adjust over time the more long-term and task independent functions, such as codes of conduct or project values, in response to conflicting norms, a lack of clarity, or the need to develop norms that do not yet exist. At the same time, Study 1 highlights the importance of the awareness of conflict, its interpretation by team members, and the role of group conflict norms that moderate the impact of conflict on eventual team outcomes.

Study 2 aimed to generalize the findings from the first study through a wide survey of FOSS projects and participants. Addressing the first and second overall research aims, Study 2 quantitatively explored the relationship between conflict processes and structural variables (interdependence, leadership style, distribution of decision-making and geographical distribution), as well as emergent states (performance perception and identification), and outcomes (intention to remain in the project). Study 2 also confirmed the 4-factor structure of conflict, and validated a set of measurement tools.

Previous chapters have already discussed more specific implications of findings from Study 1 and Study 2. This chapter will integrate the findings from both studies, and discuss them in the context of the three broader research ques-

tions and their overall contributions to work on conflict across different spaces. The following sections, where appropriate, will also highlight areas of interest for future work.

6.2 Contributions to FOSS literature

One of the goals of this dissertation was to contribute to research on the role of conflict in peer production systems, such as Free and Open Source Software development teams. Literature thus far has taken important steps to show that conflict is a commonly occurring process in FOSS teams (Weber, 2005), and described individual episodes of conflict occurrence (Elliott & Scacchi, 1993, 2003). Related work on Wikipedia has adopted the classic intragroup conflict taxonomy to classify different types of conflict episodes and measure their relative impact on team outcomes (Arazy et al., 2011, 2013). Thus the first aim of this dissertation was to bring together these diverse research threads into one conflict framework by exploring conflict manifestations in FOSS while taking into account the long-term orientation of the group and expectation of future interaction.

6.2.1 Conflict Manifestation

The findings show that conflict manifests in somewhat different but related ways to the taxonomy created by Jehn for traditional organizational teams. Specifically, the present study identifies 4 distinct conflict types, rather than 3. The first three map toward Jehn's (1997) original taxonomy – task, process and affective conflict. The final conflict type, normative conflict, appears to be distinct a feature of ongoing virtual teams. Their manifestations shall be examined in turn, with reference to relevant literature on organizational teams.

Study 1 finds that FOSS teams frequently disagree about issues related to the task at hand, that is the work of producing software. Task conflict fre-

quently manifests in disagreements about the overall project road-map, that is what features should be included in the future. Task conflict may also involve implementation debates, or conflicts that emerge when choosing among the various technical options for realizing project goals. Finally task conflict may involve decisions concerning dependencies (typically other projects and libraries), which may compete in function. At the same time, task conflict is often part of the group norms in FOSS teams, and encouraged either directly through request for comment procedures in the team or implied in practices such as code review. This manifestation is fairly similar to the original conceptualization by Jehn (1997), and highlights in particular the recognition by teams of the positive impact of task conflict that needs to be encouraged (Amason, 1996; Jehn, 1995; Pondy, 1967). (Later portions of this chapter will expand further on the connections between conflict types and outcomes).

Process conflict reported by FOSS teams was largely logistic in nature. Specifically, procedural issues involved disagreements about how to write software more generally, such as the appropriate style to use. Style is a common source of disagreement for new contributors who are not yet familiar with the way things are done in a project, as well as between established team members who have strong convictions concerning their individual approaches. Participants point out that issues concerning contribution, that is, task assignments and how well they are being carried out, are relatively rare because the projects are voluntary endeavors and thus it is different to hold members accountable for inactivity. However, there are occasional debates concerning giving credit, attribution and respecting hierarchy. This manifestation of process conflict is somewhat different from that of traditional teams in that it largely concerns logistical issues, while traditional teams are more likely to debate issues of contribution and responsibility (Behfar et al., 2011; Jehn, 1997). This manifestation also points to the salience of separating process conflict into two distinct dimensions of logistics and contribution (Behfar et al., 2011), as the above findings

show the two dimensions may exist in different proportions in teams of different structures and motivations.

Affective conflict manifestation is one of the more interesting types of conflicts to be examined in this study in part because it did not appear to be reported in isolation by participants in Study 1. Interpersonal disagreements and flames typically emerged on the foundations of earlier and other kinds of conflicts. This is not unexpected as conflict episodes have been known to transform into affective conflict over time (Arazy et al., 2013; Greer et al., 2008; Jehn, 1995). What is interesting is that unlike findings in traditional teams, affective conflict does not appear to be reported as arising on its own. This may be a feature of the high virtualness of FOSS teams, namely the relative invisibility of participants physical characteristics and asynchronous communication which can serve to increase group identification by minimizing apparent differences that would otherwise cause interpersonal friction (Postmes, Spears, & Lea, 1998; Walther, 1996). However, manifestations of task or procedural conflicts may serve to shed light on interpersonal differences through greater discussion, leading to affective conflicts. Furthermore, once affective conflicts do emerge on the foundation of interpersonal differences, they may supersede the collective group identity (Lea & Spears, 1991). In absence of a unifying identity these conflict episodes can become rather intense and turn into protracted flame wars because of the availability of the same communication characteristics: relative invisibility that allows more vitriolic speech, and asynchronicity that allows individuals to flame and run (Postmes et al., 1998).

Finally, this dissertation also highlights a fourth type of conflict that appears to be particularly salient to self-organizing ongoing teams like FOSS projects – normative conflict. Normative conflict manifests as disagreements about group norms that are unclear, have not yet been formalized, or by highlighting group behavior (descriptive norms) that has become inconsistent with group values (prescriptive norms). In this way normative conflict is an avenue through which

teams adjust their structures over time, and provides a check for growing teams that need to formalize structures and ensure these structures continue to make sense. Normative conflict is particularly relevant to ongoing virtual teams for two reasons. First, due to their ongoing nature, they have a greater tendency toward evolving group norms (T. Ghosh et al., 2004; Postmes et al., 2000). Second, their greater virtuality results in a tendency to overcome uncertainty by archiving activities and explicitly stating group norms (Bolici et al., 2009; Dabbish, Stuart, et al., 2012). Thus the ongoing virtual context affords greater opportunities for observing inconsistencies in expected and actual behavior, as well as the clarification of norms. Normative conflict is therefore an important addition to the intragroup taxonomy that is relevant for the study of ongoing virtual teams. A separate section on normative conflict below will elaborate on these findings and their implications for broader research.

The present dissertation uncovers the above four-factor structure in the first qualitative study, and validates this structure in the second study across a variety of FOSS teams. In doing so, the present work contributes a reliable, consistent and relevant conflict taxonomy that can be used by future researchers on open collaborative systems to examine additional factors that may be relevant to conflict occurrence in FOSS, such as conflict management styles, additional structural variables, emergent states or outcomes. Using a consistent framework affords comparison across different studies to collectively build a rich and complex understanding of conflict. Furthermore, a framework built on existing conflict theory in traditional organizations also affords comparisons across different organizational types. This chapter will utilize this to contrast findings between FOSS teams as examples of ongoing teams, short-term virtual teams and collocated teams below.

At the same time, as the next section shows, the present work highlights the highly subjective nature of conflict, with the same issues and conflict types potentially interpreted in different ways by the teams themselves, leading to

potentially different outcomes.

6.2.2 Conflict Interpretation and Conflict Norms

A related aim of this work was to understand the different ways in which the conflict types above were interpreted in the context of FOSS development teams. The awareness of conflict and its attribution are critical steps that determine whether subsequent team member behavior concerning the issue is constructive or destructive (Pondy, 1967; Thomas & Schmidt, 1976; Thomas, 1992; J. A. Wall & Callister, 1995). For instance, Pondy (1967) argues that when factors come together that frustrate one or more parties, but are not perceived to be conflicting by team members, conflict may remain dormant (in a 'latent' stage). Thus before conflict can emerge within a team, the episode must first be recognized as a frustrating force by one or more team members (Thomas & Schmidt, 1976). Specifically, before conflict manifests in a negative way within the team, it must be felt to be negative by at least one party. At the same time, the interpretation of conflict as negative is the result of the attribution of another party's behavioral intent as negative (whether or not this is actually the case) (Thomas & Pondy, 1977). As Jehn (1997) has found, group norms surrounding conflict are one source of conflict interpretation and attribution, thereby determining when an episode is seen as conflict by team members.

Study 1 highlighted that what is considered conflict in FOSS teams can vary quite dramatically across different projects. Many teams have a pro-task conflict norm that is either explicitly written into the contribution procedure (such as a request for comment), or implied in the course of making a commit (such as the code review that typically accompanies a submitted patch). This sort of institutionalization of conflict into group norms provides an important stabilizing mechanism that ameliorates some of the negative effects of conflict by suggesting a positive interpretation of the conflict episode (Coser, 1957).

Other projects borrow conflict norms from their leaders or benevolent dicta-

tors. For instance, Chapter 3 highlights how the Ruby community is relatively conflict averse because "Matz is nice", but in the Linux community Linus Torvalds strongly defends his use of affective conflict and flaming on developers as a way of accomplishing things in the project (Sharp, 2013). Thus the norms that become prominent in initial stages of group development, such as those emerging from the influence of founder's personalities, appear to have a persistent and strong influence on subsequent team behavior. Feldman (1984) describes this process in the evolution of collocated teams and terms it "primacy".

Study 2 initially included a measure for conflict norms based on scale validated by Jehn (1995). However, while the scale showed reasonable performance during pre-tests, reliability was unfortunately not sufficient during the final data collection to be used in the study. This is a methodological limitation of the current work, and future work is strongly encouraged to reexamine this value with a reliable scale for the measurement of pro-conflict norms, thereby building on the above initial qualitative findings.

Taken together, the subjective nature of conflict in FOSS settings has important implications for research on the phenomenon. Specifically, while all FOSS teams may experience the above four conflict dimensions, the extent to which these dimensions are interpreted as destructive or necessary aspects of team collaboration are firmly rooted in project norms. It is thus imperative to control for this element in future work that aims to examine conflict in relation to FOSS team emergent states and outcomes.

The rest of this chapter is laid out as follows. First, the chapter examines in greater detail the emergent normative conflict dimension, and considers its implications for research on intragroup conflict literature more generally. Then, the chapter contrasts findings on ongoing virtual teams with short-term virtual and collocated teams in traditional organizational settings, to build a more general understanding of conflict as a team process. The chapter then concludes by briefly re-stating the theoretical and practical implications of conflict an-

tecedents in ongoing virtual teams like FOSS.

6.3 Contributions and Implications of the Normative Conflict dimension

By situating the study in an ongoing virtual team context, the present findings highlight a relevant addition to the intragroup conflict taxonomy in the form of normative conflict. This research therefore contributes to both work on virtual teams and intragroup conflict in work groups in general, by illustrating, first, the way normative conflict impacts the formation of team norms, and second, its role in the subsequent evolution of team structure. In explicating these mechanisms, this dissertation is addressing two important research gaps identified in a recent review of virtual team literature by Martins and Schilpzand (2011) as well as speaking to the growing body of work investigating non-linear team processes and cycles more generally.

Findings show that normative conflict has an overall negative impact on team emergent states and outcomes. Normative conflict is negatively connected with both performance perception and intention to remain in the project by acting indirectly on identification. Normative conflict also interacts with task conflict such that in the absence of productive task debates, normative conflict has a much more negative impact on team performance perception. Thus normative conflict emerges as an important dimension to study and manage in ongoing virtual teams such as FOSS teams.

The idea of normative conflict idea is not new and was proposed by Packer (2007) to explain the possibility of dissent in highly cohesive groups, as well as appearing in several other forms across different fields (as reviewed in Chapter 4). To recap, Packer finds that when individuals identify strongly with a team, they are more likely to speak up if they believe their observations of the teams' behavior are not in line with expected behavior. However, individuals

with lower identification levels are more likely to disengage from the group entirely. The present study highlights an interesting complement to Packer's work. First, the present work finds normative conflict also emerges out of unclear behavioral standards, and not just a dissonance between team actions and prescribed behavior. And second, Study 2 highlights that normative conflict has a negative impact on identification with the team, and thereby indirectly reduces the intention of ongoing virtual team members to remain in the project.

Crucially, while in Packer's conceptualization the relationship between normative conflict, identification and retention is positive, in the present study, normative conflict has a negative relationship with identification, and through identification, it also a negative relationship with intent to remain. Furthermore, in Packer's work, identification is an input factor that is manipulated in the study, while in the present work identification is treated as an emergent state that constantly evolves due to team processes (Marks et al., 2001).

Taken together, this contradiction may highlight an interesting and complex long-term mechanism of normative conflict action. While individuals who strongly identify with the team are more likely to initiate normative conflict episodes (Packer, 2007; Packer & Chasteen, 2010), this may have an overall negative effect on the identification of other team members by exposing faultlines within the group and divisions along certain issues that were not previously apparent (Lau & Murnighan, 2005). Thus while the expression of normative conflict may come from a good intention toward clarifying group norms, it may in extreme cases splinter the community. In the case of FOSS projects, this may lead to a project fork, as the normative conflict example in the Node.JS community from Study 1 documents.

As far as this author is aware, this is the first work to bring the idea of normative conflict into the space of organizational group research, thereby expanding the intragroup conflict taxonomy. However, indirect evidence for a process akin to normative conflict has already existed across different strands of research on

conflict in organizational teams. For instance, a number of classic group development theories viewed conflict as part of the team life cycle. Tuckman (1965) argued that a "storming" (or conflict), phase is necessary before achieving the "norming" phase in which teams develop a shared understanding about working together. Similarly, Arrow and colleagues (Arrow et al., 2004) viewed conflict as one of primary forces for change in the team, while Poole, Van De Ven, Dooley, and Holmes (2000) suggested that among the four distinct motors for generating change in a team, there is a dialectical component that sees change emerging as the result of conflict between opposing entities.

However when the intragroup conflict taxonomy was developed and refined by Jehn (1995, 1997), it did not include a form of conflict that dealt more broadly with its force for change, and focused instead on conflict concerning the action of working together or as the result of mutual incompatibility. In other words, Jehn's conflict framework had a more teleological conceptualization of group development that assumed teams move towards a specific end goal and conflict was either instrumental to or detracting from this goal (Poole et al., 2000). On the other hand, the present study contributes an additional dialectical and evolutionary perspective that shows conflict occurs as the result of repeated cycles of opposing forces that drive team development forward. Perhaps a core difference of this study lies in its focus on ongoing self-organizing teams that have more capacity to debate and negotiate their own means of production and social structures overtime (Kelty, 2008). Yet as more teams move towards decentralization, this feature of work groups is gaining more attention (Maynard, Mathieu, Rapp, & Gilson, 2012; Tannenbaum et al., 2012).

Thus, the present study also contributes in closing the gap between early conflict research that saw conflict as a necessary step in group development, and intragroup conflict research from the previous two decades in two ways. First, the present work illustrates that manifest normative conflict is a key process through which norms are developed in ongoing teams because it helps to raise

to the forefront, and affords the opportunity to resolve, unclear or misspecified group norms. For instance, Chapter 3 documented several examples of normative conflicts resulting in the formation of a team code of conduct. In this way, the findings speak to early work on conflict in collocated teams, such as that by Opp (1982) who argued teams may form norms through explicit/voluntary and active discussion. Similarly, Coser (1957) argued that conflict helps to revitalize existent norms and contributes to the emergence of new norms. This study goes one step further to suggest discussion about norms is not always a smooth process, with competing points of view that emerge through a specific dimension of conflict – normative conflict. In this way, just like conflict can be a part of the existing discursive norms of a group, or the way a group communicates (Choi & Schnurr, 2013), so conflict can serve as the channel for crystalizing, debating and evaluating these same group norms as the team develops over time. Normative conflict is therefore consistent with predictions by Curşeu (2006) that the emergence of conflict in ongoing virtual teams would have an impact on and be influenced by the other processes and emergent states, including conflict itself.

The relationship between normative conflict and the formation of group norms is further relevant to work by Ayoko et al. (2012). They find that, consistent with Tuckman's (1965) prediction, norm setting is prompted by conflict episodes and occurs in response to conflict as stimulus. However they find this process does not occur linearly, rather it occurs at various times throughout the life of the project team. One concrete manifestation in Ayoko and colleagues' study was the development by several teams of a feedback-requesting norm to ensure openness to input, demonstration of mutual respect and that team members remain on the same page about their task interpretation. This spontaneous generation of a feedback norm is similar to the request for comment or code review norms that evolved in FOSS projects that are designed to promote task conflict as an acceptable part of team development (Chapter 3).

Second, normative conflict also emerges as a mechanism that directly trig-

gers changes in team structure beyond its indirect impact via emergent states and outcomes. In this way the present work addresses a second gap in virtual team literature, illustrating feedback processes in teams that serve as inputs for future processes or team structure (Martins & Schilpzand, 2011). One way normative conflict does this, as explained above, is through the debate and crystallization of group norms. However, normative conflict may also result in broader changes to the team's technical structure. For instance, Chapter 3 documented an ongoing debate in the Linux community concerning modularization of the code base to reduce conflict due to interdependence. This finding is consistent with work on collocated ongoing teams, such as Langfred (2007) who showed that following conflict, teams may restructure for more loose interdependence and/or less individual autonomy.

Normative conflict may also impact team social structure by encouraging the formation of coalitions within the team. For instance, Chapter 3 documented how normative conflict concerning gender-neutral language in Node.JS sparked a greater debate concerning the role of for-profit companies in the running of the project, thereby splitting the community in half around an ideological faultline. One subgroup felt strongly about the need for the open source project to be run free from company interference, while another coalition formed around the belief that doing so would be destructive to the existing ecosystem. The coalition supporting self-determination and freedom from corporate influence found sufficient support to establish a fork of the project (IO.JS). Reconciliation between two projects only came when this fundamental distinction, or faultline between the two sides, was resolved by setting up of a foundation to run the Node.JS project, reinforcing the overall group identity and allowing the two communities to be integrated into one whole.

These findings speak to an existing branch of virtual team literature concerning subgroup formation. Specifically, Lau and Murnighan (2005) found that faultlines might form in groups that are either physically or psychologically

divided by geographical location, ideology or some other distinct characteristic, into smaller sub-groups. Virtual teams that are highly geographically distributed or diverse are particularly susceptible to these effects (O'Leary & Mortensen, 2010). Similarly, Coser (1957) argues that conflicts with some team members may produce new associations or coalitions with others. In other words, the manifestation of normative conflict may make opaque distinctions of ideology and values that would not have been otherwise visible, leading to the formation of subgroups and faultlines and leading to changes in social structure. Thus, intragroup conflict in ongoing virtual teams not only emerges from both technological and social issues, but through normative conflict, it also serves to impact both technological and social structures (DeSanctis & Poole, 1994).

In summary, by introducing normative conflict as an additional dimension to the intragroup conflict taxonomy, the present dissertation makes a theoretical contribution toward our understanding of conflict in teamwork more broadly. Specifically, normative conflict emerges as one of the processes through which group norms are created, debated and continually modified. Normative conflict also influences both technical group structure and subsequent group inputs, as well as social group structure through the possible formation of subgroups and coalitions when highlighting ideological divisions. In this way, normative conflict helps address gaps in research on how ongoing team processes like conflict form feedback loops and influence not only team emergent states and outcomes, but also feed into subsequent iterations of teamwork and taskwork (Marks et al., 2001).

6.4 Contributions to Virtual Team Literature

The question of what conflict types affect team outcomes, and under which conditions, continues to be salient for research on virtual work. The present findings contribute another perspective to this debate by examining conflict specifically

within ongoing virtual teams. This section presents overall findings, and contrasts them with trends in virtual team literature in the areas of conflict outcomes and emergent states, followed by structural inputs and other moderating factors.

6.4.1 Conflict, Emergent states and Outcomes in Ongoing Virtual Teams

Process conflict

Firstly, findings from the present dissertation show an overall negative impact of process conflict on ongoing team emergent states and outcomes. For instance, Study 1 documents the way style disagreements can become a hurdle for new team members, as well as taking attention from the task being performed when two or more experienced members with sharply different ideas disagree about how software should be written. Study 2 similarly finds process conflict to have an overall negative effect on performance perception when controlling for the effect of other conflict types (although this is only significant at the $p < .10$ level). Process conflict further interacts with affective conflict such that the negative effects of process conflict are higher when levels of affective conflict are lower. This overall negative impact is consistent with prior work on both collocated and virtual that show largely negative effects of process conflict on team performance, satisfaction and coordination teams (Behfar et al., 2011; de Wit et al., 2012). This finding is also consistent across both short-term and ongoing teams (Jong et al., 2008).

Taken together, process conflict appears to have a universally negative relationship with performance, across different types of teams and organizational structures. In short-term teams, small amounts of process conflict are useful in clarifying procedures at team inception, but quickly become distracting from team outcomes in conditions of time pressure. Ongoing teams, on the other hand, have a tendency toward process conflict due to their inherent greater focus

on improving team efficiency and processes (Saunders & Ahuja, 2006). Thus process conflict in this case also takes away from outputs, but this may be true only in the short to medium-term. In the long-term, if the conflict is resolved, it may create opportunities for more efficient teamwork by clarifying misunderstandings. It would therefore be very interesting to examine process conflict in ongoing teams longitudinally, to trace its impact across different task iterations.

Affective conflict

On the other hand, Study 2 finds affective conflict to have a positive direct effect on both performance perception and identification, in sharp contrast to predictions and empirical evidence from literature on both collocated and virtual, short-term and ongoing teams (e.g. Curşeu, 2006; De Dreu & Weingart, 2003; Hinds & Bailey, 2003; Jehn, 1995; Kankanhalli et al., 2007; Lira et al., 2008; de Wit et al., 2012, among others).

It is possible this finding is a result of a methodological issue, in particular with regards to sampling. Despite the stratified random sample used, participation in the study is optional. Participants who have overall more positive experiences with their team might be more likely to respond to the survey, because those who have experienced conflict and do not identify strongly with the team may have already left the group (Furumo, 2008). However, other conflict types did emerge that show a negative relationship with team identification and subsequent intention to leave the team, suggesting an adequate sample. Furthermore, the effect sizes of the relationships between emergent states and affective conflict are substantial, and are present even in subsequent regressions that controlled for a number of other factors, such as tenure, team size, gender, role in the project, among others. Thus, another possible reason for this relatively persistent and contrasting finding may lie in the analysis. Study 1 suggested that affective conflict rarely exists in isolation in the ongoing teams studied. Therefore the individual effect of affective conflict when controlling for the effects

of other conflict types may need to be interpreted with caution if in practice affective conflict only emerges as an escalating mechanism.

Alternatively, the positive relationship between affective conflict and emergent states may suggest the presence of a mediator variable that is not specified in the current model. Specifically, as the discussion in Chapter 5 proposed, based on work by Dabbish, Stuart, et al. (2012), affective conflict may be an indirect signal of team activity level, which is in turn connected with both more positive perceptions of the team performance, and stronger identification with the team. In other words, team members who fight more, care more. Thus affective conflict may have a direct negative effect on performance perception and identification with the team, but a positive relationship with perceptions of team activity, and an indirect positive effect on these emergent states.

Furthermore, there may be an unexamined temporal effect present. For instance, prior work has shown that teams with greater shared identity experienced less negative effects of affective conflict on team emergent states and outcomes (Han & Harms, 2010; Hinds & Mortensen, 2005; Mortensen & Hinds, 2001). The work treated identification with the team as an input variable, mediated by processes like conflict. By contrast, the present study conceptualized identification with the team as an emergent state, one that may also be influenced by team processes and consistent with recommendations by Marks et al. (2001) in their temporally sensitive taxonomy of team processes. Taken together with prior work, present findings may suggest identification and conflict are engaged in a complex feedback loop in which overall group identification is an emergent outcome that serves as future input for other conflict processes. Thus, even if affective conflict is related with negative perceptions of performance and reduced identification at the point in time the conflict erupts, by signaling greater activity in the team it serves as a positive signal in the long term, especially for those not directly involved in the debate. It would therefore be crucial for future work on conflict in ongoing virtual teams to trace this feedback process of conflict on

identification with the team over time.

Task Conflict

Task conflict's effect on team emergent states and outcomes has notoriously been conflicting in research findings across both virtual and collocated teams (de Wit et al., 2012). In the present study, though task conflict was the most frequently reported type of conflict, no significant direct or curvilinear effects were found between task conflict and either performance perception or identification with the team. Taken in the context of findings from Study 1, this pattern makes sense as participants often see task conflict, or criticism of proposed ideas, as part of their workflow when contributing to FOSS projects. Interestingly, task conflict interacted with normative conflict such that normative conflict had a less negative impact on performance perception when task conflict levels reported were relatively higher. In other words high task conflict levels in ongoing virtual teams, particularly those with a positive social norm toward it, may be helpful in keeping the team focused on their goals and minimizing unproductive normative conflict.

This finding is somewhat consistent with predictions by Curşeu (2006) for ongoing virtual teams, who suggested task conflict may have a positive impact on the team under the right conditions. This is also consistent with recent work on Wikipedia by Arazy and colleagues (2011, 2013) who found that at higher levels of task conflict, cognitive diversity had a positive impact on quality output, and that task conflict did not significantly impact article quality if it did not occur together with affective or process conflict. This is also consistent with the meta-analysis by de Wit et al. (2012) who found no significant correlation between task conflict and performance across both collocated and virtual teams more broadly.

On the other hand, the present findings stand in contrast to some earlier work on short-term virtual teams that found task conflict to be negatively related with

team potency (Lira et al., 2008). This may be because short-term student teams have less time to develop group norms that favor feedback and criticism, or because in more short-term oriented teamwork this type of norm setting activity is less productive given the limited time available to accomplish a task.

Summary

Taken together, the above findings on the four conflict dimensions and their impact on emergent states and outcomes in ongoing virtual teams show some intriguing differences when contrasted with short-term virtual teams and collocated teams. On one hand, ongoing teams have more time to develop group norms about conflict, such as pro-task conflict norms that enable cognitive diversity and minimize the misinterpretation of criticism as personal. On the other hand, this same long-term orientation and ability to evolve group norms may lead to normative conflict when these group norms are unclear or inconsistent with everyday practice. This process is connected with a reduction in identification with the team, and greater intentions to leave the project. In fact, normative conflict emerges as the sole predictor of intention to remain among the four conflict dimensions, as Chapter 5 showed. Furthermore, in ongoing teams, especially groups that rest on voluntary efforts of participants like FOSS projects, affective conflict appears to be a signal of a healthy community rather than a performance distraction or motivation to leave the team. Therefore, the findings show that it is normative and process conflict, rather than affective or task, that require special attention and management in ongoing virtual teams. The next section will briefly examine the interrelationships between the various conflict types uncovered, and their relevance to broader literature.

6.4.2 Conflict Transformation

The present findings extend a growing body of work that suggests different kinds of conflict may be interrelated and transform into each other over time (Arazy et

al., 2013; Greer et al., 2008; Jehn, 1995; Mooney et al., 2007). Specifically, the two studies presented in this dissertation highlight a number of possible transformations that reflect findings in earlier work, such as task into affective conflict. Even early work using the intragroup conflict taxonomy had noted that a small percentage of task conflict episodes had transformed into affective conflict due to a misattribution of intentions (Jehn, 1995). Mooney et al. (2007) examined this relationship more formally and found that task conflict mediates the relationship between affective conflict and team inputs. Study 1 similarly finds that both task and process conflicts can gain an affective dimension over time. The present findings therefore lend further weight to the assertion that affective conflict may begin as either task or process conflicts that have transformed due to misattribution and social judgment.

As already mentioned above, affective conflict does not appear in isolation in the qualitative findings of Chapter 3. While an absence of observations is not conclusive evidence that the pattern does not exist, taken together with the above work on conflict transformation, the pattern suggests affective conflict may commonly occur as a form of conflict escalation rather than arising directly out of team inputs or task characteristics. In support of this, early work by Glasl (1982) had noted that, if left unchecked, conflict can continue increasing in intensity.

At the same time, more complex interrelationships can exist within the same conflict episode, such as between affective, process and normative conflict, as observed in the Rails and GitHub hacking issue, described in Chapter 3. To recap, the conflict began with a normative dissonance between community norms about convention over configuration and a potential security flaw, and then gained an affective dimension through the frustration of the developer who reported the issue for not being heard. The conflict also gained a procedural dimension when the developer violated established norms for contribution, and hacked into the project to demonstrate the issue severity. Though earlier

work has not directly traced the evolution of one conflict episode across more than two forms of conflict, a longitudinal study design by Greer and colleagues (2008) showed process conflict can be a predictor of different task and affective conflicts in the future across several time periods, as well as more process conflict.

Taken together, these findings suggest conflict can transform in a variety of different ways and this has significant implications for future research. More specifically, the variety of possible conflict transformations problematize studying their individual impact on the various team emergent states and outcomes that research needs to consider. For instance, Chapter 5 highlighted the significant impact of the process-affective and task-normative combinations on performance perception. Arazy et al. (2013), on the other found that both task-affective or task-process conflict combinations were negatively associated with quality of output. It quickly becomes impractical to examine the effect of all kinds of conflict combinations on the various significant emergent states and team outcomes under an equally complex variety of team structural inputs.

It may instead be more illuminating for future work to focus on identifying boundary conditions under which conflict transforms into another type. Defining more theory around boundary conditions may help in quantifying the extent of conflict escalation, and facilitate more targeted conflict detection and resolution. It may also shift the way we investigate conflict effects – the immediate transformation of, for instance, task into affective conflict may have a short-term impact on the team output because this takes attention away from performing the task. However its long-term impact may depend on a number of other factors such as whether and how the conflict is resolved, and/or what other type of conflict the episode evolves into. As such, for studies that address the time dimension in conflict research, it may also be relevant to consider not just the type of conflicts involved, but their intensity over time, and the total time spent on a particular conflict episode as a whole because this may help to shed more

light on output implications.

In fact, research on collocated teams is beginning to move in this direction. A recent study by Ayoko and colleagues (2012) drew on Tuckman's (1965) model of group development to trace conflict episodes, their transformation and escalation over time in face-to-face university student teams They found that teams who did not leave the "forming stage" and spent over a third of their time in the "storming" (conflict) stage performed most poorly on the given task. Importantly, the team that scored the best dispensed with storming (conflict) much earlier on in their group life cycle than other teams, and spent the largest portion of their time in the "norming" and "performing" stages. Their findings show that the time spent in conflict is as important to success as its eventual resolution. Their findings are also some of the first to highlight the boundary conditions under which conflict episodes transform. For instance, their work shows that during the "storming" phase, perceived social loafing of team members moves task conflicts into an affective dimension, while during the "forming" stage, affective conflict is connected with personal feelings of frustration in not understanding the nature of the task given and the inability to make progress in task ideation. In other words, different boundary conditions may result in conflict transformations at different stages of a group's development suggesting more longitudinal research of ongoing virtual teams is necessary.

Taken together, the multitude of conflict transformations uncovered in this study and prior work on conflict in ongoing virtual and collocated teams suggest a need to shift the focus toward theorizing broader relationships rather than focusing on exact conflict transformation combinations, as well as employing different measurement strategies. In particular, work on collocated teams suggest that examining boundary conditions of conflict transformations at different stages of the group development, as well as the length of time spent on conflict may be useful avenues for future research.

The next section will explore factors that give rise to the different forms of

conflict, and once more connect this with overall patterns in literature on virtual work.

6.4.3 Conflict Antecedents

Qualitative findings in Chapter 3 showed four distinct structural factors that may work to increase or reduce conflict in ongoing voluntary virtual teams, while quantitative findings from Chapter 5 showed support for the assertion that different independent factors are likely to trigger different kinds of conflict (Curşeu, 2006). Chapters 3 and 5 have already discussed these findings and their implications in some detail. Therefore, the following final section contrasts findings across both studies and discusses their implications for the overall research aims.

Study 1 found that bikeshedding (Kamp, 2003) emerged due to a combination of greater virtualness in FOSS teams as well as the relative empowerment of team members to participate in the decision-making process. Specifically, the greater transparency of communication and tendency to archive group interactions affords participation in conflict episodes for other team members not directly connected with the conflict episode, as well as the general public. As a result, discussions may escalate due to the introduction of new ideas and opinions. These discussions may also crystalize existing faultlines in the community as more individuals join the discussion and form coalitions. The effect is particularly strong for more simple issues, because these are more accessible to the general team, than complex problems that may benefit from more points of view. Thus bikeshedding can take attention away from more complex tasks by focusing too much attention on trivial issues.

At the same time, Study 2 found that more distributed decision-making was associated with lower process and normative conflicts, the two conflict types with negative impacts on team performance perception, identification and developers' sustained participation. More distributed decision-making was also directly connected with a greater sense of identification with the team.

Thus greater distribution of decision-making in ongoing virtual teams has a complex and multi-faceted relationship with conflict emergence. On one hand, in combination with greater communication transparency, it leads to more protracted conflicts that involve more community members. On the other, it can facilitate a reduction in process and normative issues. It would be instructive for future research to examine the specific mechanisms through which distributed decision-making has this differential effect on conflict emergence, in order to inform community design. Specifically, are there potential mediators that can help reduce the effect of bikeshedding in transparent virtual environments?

Second, findings from both studies highlight the importance of leadership in bridging gaps in mutual understanding when dealing with different kinds of manifested conflict episodes. Specifically, Study 1 highlights the way different individuals may experience "different truths" concerning the same episode – for instance, when giving critical feedback on someone's work, one party may see this as a purely task related disagreement concerning implementation, but for another party this may be interpreted as a personal criticism and therefore an affective conflict. Study 1 findings further suggest that an authority figure or developer with a reasonable standing in the community can serve to reduce conflict levels. Study 2 finds support for this assertion. Specifically, transformational leadership exerts both a direct positive effect on team performance perception, as well as indirectly improving performance perception by reducing normative conflict levels. Furthermore, transformational leadership also improves team identification and intention to remain in the project by indirectly by reducing normative conflicts. Thus leaders are able bridge the "different truths" by serving as mediators in disagreements, as well as stepping in to clarify normative inconsistencies.

6.5 Conclusion

The present dissertation contributes to our understanding of conflict in FOSS teams in a few ways. In mapping out different kinds of conflict experiences in FOSS teams, the work contributes a relevant and consistent model of conflict that can be used by future research into the FOSS phenomenon as well as to contrasting findings between ongoing virtual teams and other forms of organization. In doing so, the work proposes and validates the addition of normative conflict to the intragroup taxonomy as a dimension particularly relevant for ongoing virtual teams. Finally, through normative conflict, the dissertation describes how conflict affects ongoing group development through the clarification and establishment of group norms, as well as in serving as an input for future structures and interactions.

Chapter 7

Conclusion

The present study set out to develop a unified and relevant framework for understanding conflict in the context of peer production systems like Free and Open Source Software development. Previous research has begun to examine conflict, but at present, the body of work lacks a consistent framework to allow comparison across different studies (Crowston et al., 2012). It is important to understand how conflict is manifested and its effects on the team outcomes and long-term development because FOSS projects represent a critical ecosystem supporting many important Internet applications. To begin doing so, we need a consistent set of tools that allow researchers to build more complex models and test them in collaboration across different contexts.

Furthermore, by situating the study in the FOSS context, the present dissertation was interested to understand conflict in ongoing virtual team settings, and in particular whether its nature and manifestation would differ from short-term virtual teams and collocated teams. In doing so, the study aimed to fill a gap in general literature on virtual work that has thus far examined processes in temporary virtual teams with fixed membership structures and distinct end points (Gilson et al., 2014; Martins & Schilpzand, 2011; Saunders & Ahuja, 2006). Examining processes in ongoing virtual teams is important. Due to a lack of a specified end point, ongoing teams have greater sense of history and expec-

tations of future interaction, while at the same time also needing to deal with greater change over time (Saunders & Ahuja, 2006). These conditions dramatically alter the focus of teams towards more long-term goals like efficiency, team well-being and member support. These conditions also better represent the nature of virtual work today compared to temporary distributed groups (Maynard et al., 2012; Tannenbaum et al., 2012). Thus, in examining conflict in an ongoing virtual team setting, the present work contributes an important insight to both theory and practice of management of virtual teams.

Finally, and related to the second aim, the present work aimed to understand the impact of conflict on long-term team structures and team development in ongoing virtual teams, addressing a gap in research identified by Martins and Schilpzand (2011). Due to their long-term focus, ongoing teams may develop and evolve structures based on their experiences and disagreements with one another. This is an important gap to address because it shifts the focus of conflict research from a teleological one concerned with supporting or detracting from team outcomes, towards a more evolutionary perspective that recognizes conflict as an inevitable part of the team life cycle.

Thus the present study aimed to address the following three related research questions:

1. How is conflict manifested and interpreted by FOSS project members? What impact does this have on the relationship between conflict, team structures, emergent states and team outcomes?
2. How does conflict as a process differ in dynamic ongoing virtual groups like FOSS projects compared to conflict in short-term virtual teams in traditional organizations?
3. How does conflict impact structural change in dynamic and ongoing virtual groups like FOSS projects?

This final chapter will conclude by briefly restating a synthesis of the empirical findings across the two studies reported in the dissertation, exploring how

they fit together and relating them to the above research questions. Next, theoretical implications of the findings will be restated, in particular with reference to how these findings speak to the broader body of work on conflict in organizations. Following this, the chapter will summarize practical implications for both FOSS community leaders and ongoing virtual team managers. The chapter will then cover methodological limitations, exploring how the empirical studies reinforced each other's limitations, future research plans and recommendations for future research directions, before concluding with a restatement of the central argument.

7.1 Synthesis of Empirical Findings

Addressing the first research aim, the present work identifies four distinct kinds of conflict experienced by FOSS teams, then reformulates and validates them in an updated conflict framework for ongoing virtual teams.

- **Task Conflict.** Findings suggest the most commonly occurring disagreements involve conflict about the task to be performed, that is, the creation of software. Task conflict includes issues such as project road-map, feature selection and prioritization, as well as decisions about competing technical implementations and dependencies. Task disagreements are often part of the written or unwritten group norms concerning working together, and are often encouraged through social processes like formal requests for comment or code reviews.
- **Process Conflict.** Teams also disagree about procedural issues, mainly concerning logistics of creating software such as the general coding style to use. Procedural issues may also arise when an individual's authority over an aspect of the project is disrespected.
- **Affective Conflict.** Task and Procedural issues may gain an affective, or emotional dimension, sometimes leading to intense disagreements, if

conflicts are misinterpreted as a personal attack or an episode is left unresolved.

- **Normative Conflict.** FOSS teams, being examples of ongoing virtual teams, also experience conflicts concerning overall group norms, values and mission. These normative conflicts occur when group norms are unclear, do not exist, or when prescribed group norms are not in line with the every day actions of the project members. In this way, normative conflict leads to the debate and evolution of new group norms, as well as changes in both social and technical structures.

Furthermore, the present study finds that these four conflict types are highly subjective, with the same episode potentially being interpreted in different ways by participants. Interpretation can vary due to personal differences and perspectives, individual relationships with members of the group, as well as between teams due to different norms surrounding conflict. Thus the findings highlight the various forces involved in sense making surrounding a conflict episode in teams with a sense of history and an expectation of future interaction (Axelrod & Hamilton, 1981). In particular, a familiarity with other team members and their backgrounds can often moderate the severity of an episode, such as being more likely to accept criticism from a contributor with greater tenure and experience in the project. At the same time, group norms set expectations for future interactions at the team level, suggesting possible interpretations of conflict, such as more positive norms towards task conflict.

Addressing the second research question, the present work finds several significant differences between conflict in ongoing virtual teams, temporary virtual teams and collocated teams. Firstly, the study shows that when contrasted with findings from studies on short-term virtual teams, ongoing teams do not appear to suffer negative effects of task and affective conflict, with the emergence of process and normative issues contributing the bulk of variance in team potency and vitality. This is consistent with predictions made by Saunders and Ahuja

(2006) who argued that ongoing virtual teams would have a greater need to focus on developing efficiency, while maintaining team identification and satisfaction. Consequently, the present study finds conflicts emerging from these issues are the most critical to the long-term success of the group, with task conflicts being more a part of group norms and affective conflicts an indicator of attachment and strong identification in the group. Furthermore, the present study highlights normative conflict as a new element in the conflict taxonomy, and a distinct feature of ongoing teams that arises from their long-term orientation and greater focus on improving processes, member satisfaction and team identification, while needing to adapt to change.

Ongoing virtual teams like FOSS teams also differ from ongoing unmediated teams in their ability to easily and automatically archive their computer-mediated interactions, and this has implications for our understanding of conflict. Past research has found that archiving affords virtual teams more transparent communication because it increases a sense of context, allowing the whole team to witness interactions and participate in decisions made (Dabbish, Farzan, et al., 2012). The present work shows that alongside these positive benefits, the greater degree of openness in virtual communication may afford greater visibility of conflict episodes to the entire group. In FOSS teams, who frequently rely on publicly available online discussions and communication archives, this may also attract attention from outside of the project. Conflict episodes may escalate due to increased attention and involvement from other parties, while coalitions may form around supporters of one or the other position (Jehn et al., 2013).

Furthermore, the weighing in of other parties is especially true for more trivial decisions and debates, as compared to more complex decisions, which is known within the community as a "bikeshedding" effect (Kamp, 2003). Thus complex issues that may benefit from more varied opinions are paradoxically less likely to attract them, while trivial issues may result in drawn out and circular debates as too many different people weigh in on the decision. It is im-

portant for future work to understand mechanisms that can help to reduce this phenomenon in virtual team settings while allowing the team to continue to benefit from the shared context created by archiving communication logs.

Addressing the third research question, the present work documents the emergence of normative conflict and its inclusion in the expanded conflict framework. Normative conflict is a key mechanism in the debate and evolution of group norms, as well as other social and technical structural changes to the group. Specifically, because normative conflict arises out of a lack of clarity or inconsistency in existing group norms, it encourages explicit debate and evaluation of these norms in order to resolve the issue. Thus normative conflict leads to the specification or reformulation and evolution of group norms.

Normative conflict can also be a catalyst for changes in both social and technical structures. Because normative conflict may surface tensions and opposing opinions from team members that concern the group as a whole, and may otherwise have been unspoken, it may serve as a catalyst for the formation of subgroups around different camps of supporters. In this way, normative conflict can fragment the social structure of a team by reducing team identification and promoting identification with coalitions instead. Thus, in addition to predictions made by Lau and Murnighan (2005) that faultlines may form during team inception due to quick self-categorization, the present work further suggests faultlines to emerge from normative disagreements even after a stable group identity has been achieved. Normative conflict can also be a catalyst for more technical group restructuring such as choosing or modifying communication and collaboration systems, or levels of interdependence in the project code base. Thus the findings also echo work by DeSanctis and Poole (1994) who argue in virtual contexts it is not only the technical conditions that give rise to different social structures, but in turn, social processes like conflict can impact the technical structures of the team.

7.2 Theoretical Implications

In addressing the first research question, the present work contributes a unified and relevant framework for studying the process of conflict in peer production settings. The framework specifies four distinct conflict types, and provides initial expectations about their mechanics in influencing emergent states, outcomes and future team structural changes. To achieve this, the work brings together knowledge gained across the different studies conducted on conflict in the open collaborative context to-date, and weaves them into an updated conflict framework. Specifically, findings reflect both the relevance of drawing on a traditional organizational conflict taxonomy (e.g. Arazy et al., 2011, 2013), as well as the need to update this for an ongoing virtual team setting. The present findings thus further speak to early work by Elliott and Scacchi (2003) whose findings suggest conflict can emerge through inconsistencies between prescribed group norms and visible group actions. In this way, the present work both supports and extends earlier work on the peer production phenomenon, providing a four-factor conflict framework that may be used to compare results across future research and build a more complex understanding of peer production processes.

In addressing the second research aim, the present work also expands theory on ongoing distributed team research by bringing theory on unmediated organizational teams into the virtual space. Specifically, the work draws on recent theoretical work on the role of conflict in team development, such as that of Jehn et al. (2013) in an unmediated context as well as classical work by Coser (1957); Gersick (1988); Tuckman (1965) and others. On this foundation, the present work builds up a conceptualization of conflict in ongoing virtual teams that has more long-term implications beyond influencing immediate team outcomes. Furthermore, the work explores the relevance of these theoretical propositions in a mediated context and delineates their boundaries. The distinction between task, process and affective conflict continues to be relevant in an ongoing virtual context and allows us to draw direct comparisons between mediated

and unmediated, short-term and ongoing work. However, the existing theory lacks a dimension that takes into account the expectation of future interaction and shifting focus of ongoing virtual teams towards efficiency, team member well being, growth and adaptation to change (Saunders & Ahuja, 2006). Thus normative conflict is proposed to extend theory on virtual teams based on both insights from literature on conflict in team development as well as that of social psychology.

By adding normative conflict to the intragroup conflict taxonomy, the present work addresses the third aim. Specifically, the work specifies a concrete mechanism for the way conflict impacts group norm development and subsequent team social and technical structures. This addresses an important gap in our understanding of virtual team dynamics by showing how normative conflict serves as an adaptation mechanism towards change and recursively impacts team inputs under the IPOI framework (Gilson et al., 2014; Martins & Schilpzand, 2011). Normative conflict brings to the forefront inconsistencies in group norms that develop over time as a result of internal development and external changes, and thus becomes a critical signaling factor for teams about potentially divisive issues. This may lead to norm reformulation, as well as technical re-appropriation, or social fragmentation if the conflict remains unresolved. The inclusion of normative conflict in the intragroup conflict taxonomy brings together diverse threads of related theories across different disciplines. The findings support Packer's (2007) work on the emergence of dissent concerning group issues and norms as a whole even in groups with strong cohesive identities. Previous research also conceptualized normative conflict to largely occur at an intraindividual level, that is, as an internal debate or conflict within an individual's consciousness. The findings therefore also extend earlier work by showing how normative conflict may manifest as an active source of disagreement that influences not just individual behavior but also group norms and long-term team structure.

Finally, the present research takes the opportunity to examine the evolution of conflict dimensions into each other in ongoing virtual teams, thereby expanding theory on conflict transformation. For instance, the causal relationship between task and affective conflict has been hypothesized and observed in prior work (Jehn, 1995; Jehn & Bendersky, 2003; Simons & Peterson, 2000; Yang & Mossholder, 2004). The present findings lend support to the theory that affective conflict can emerge as an escalation mechanism from other types of disagreements, such as task, procedural and normative issues. Prior work has also found that task conflicts may evolve procedural dimensions, an observation that is also supported by findings in this study. However, the present work further finds interactions between task and normative conflict, as well as affective and processes conflict, that have differential impacts on team emergent states and outcomes. Furthermore, some conflict episodes may evolve more than two types of conflict over the course of a protracted disagreement. Therefore, the present findings suggest the need to focus on examining boundary conditions of these transformations, such as when and how a conflict evolves an affective, or procedural, or normative dimension, rather than individual combinations of interactions. This would allow a broader understanding of conflict evolution in ongoing teams over time, as well as illuminate specific points of intervention for team managers.

A focus on more longitudinal examinations of conflict transformations may also help to address one of the paradoxes highlighted in this study concerning process conflict. Specifically, Saunders and Ahuja (2006) argue that process conflict is more likely to emerge in ongoing teams due to their increased focus on team and procedural effectiveness. At the same time, the present findings show that process conflict has a largely negative effect on team emergent states like performance perception. It may be illuminating to understand if there are differential effects of process conflict in the short-term and in the long-term. For instance, in the short-term, process conflict may indeed have a negative impact

on performance because it takes attention away from working on the task in favor of procedural discussions. On the other hand, in the long-term, addressing the issues raised in process conflict may help to improve team processes and promote efficiency.

7.3 Practical Implications

The study also offers practical advice for ongoing virtual team and online collaborative community management. In particular, the study highlights the importance of establishing clear norms about conflict within ongoing distributed teams as these may vary widely across different teams and projects. For example, depending on the focus of the project, some teams may develop an expectation of giving detailed, honest and critical feedback for contributions that is meant to generate discussion while other teams may find it more important to manage interpersonal relationships and thus be very conflict averse. The dissertation findings suggest both approaches may be successful if the teams explicitly highlight these norms and set appropriate expectations for future interactions among team members.

Furthermore, in particular during manifestations of normative and procedural conflict, it is important for team members to bridge gaps in the parties' perceptions of the conflict episode. Different parties may experience "different truths" about the same event, that is, perceive the conflict to be about qualitative different issues. An authority figure or mediator who is able to translate these different perspectives, or a community member who is able to see the issue from the other parties' point of view will be more successful at de-escalating the episode.

At the same time, it is important to identify and manage emerging normative conflict carefully. Normative conflict was the strongest predictor of team vitality, with higher levels resulting in lower identification with the team and less

intention to remain in the project. Because it can highlight potentially divisive issues, it may be productive for team leaders to work on maintaining a sense of unity and group identity in the face of potential coalitions that may form. For example, it may be useful to explicitly prioritize the overall collective mission, and work towards a resolution from a perspective that benefits the group as a whole, as this could help activate a stronger group identity (Reicher et al., 1995; Turner et al., 1987).

Finally, the present work shows that conflict has somewhat different pitfalls across the different leadership models examined in this study. For instance, on one hand, teams with flat organizational styles and distributed decision-making have more individual autonomy that reduces the occurrence of process and normative conflict. On the other, they are in more danger of engaging in long, drawn out debates due to the ready archiving of conversations that allows anyone in the team (or even outside the team, for public FOSS discussions) to weigh in. For these teams, a more formal procedure geared toward reducing multiplicity of ideas would be beneficial, such as requiring dissenters to propose an alternative solution (the Apache model), or appointing peers with the support of the rest of the group who are able to step in at times and make a difficult decision (the Debian model).

Conversely, teams with less empowerment and more prominent hierarchies are more likely to experience process and normative issues. This may be due to information overload, a lack of oversight of all elements of teamwork and a lack of power in decision-making from team members. However, findings also show that criticism from community leaders and authority sources is often more readily accepted. This is especially true when status is more visible, leaders are charismatic, lead by example, and are respected for their technical capabilities. Thus leaders may find it beneficial to work on developing a more transformational leadership style, while design of communication tools should highlight team member experiences to facilitate inference.

7.4 Limitations and Directions for Future Work

The present work has offered insights into the interpretation of an important group process by ongoing virtual groups, and was conducted through a combination of interviews, participant observation and self-report surveys. As a direct consequence of this methodology, the study encountered a number of limitations, which need to be considered.

7.4.1 Sample

Study 1, being qualitative and explorative in nature, focused on recruiting participants to address emerging questions and theory gaps while the grounded theory was being built. Thus while the study interviewed and interacted with a very diverse set of participants, this could not be a truly representative sample of the FOSS development population. To address this limitation, Study 2 recruited participants for the survey that were randomly selected based on a stratified sample derived from GitHub projects containing more than two members. Because FOSS projects with many developers are relatively uncommon (a majority are small teams or just a solo participant), stratification of the sample based on team size was first conducted to ensure equal numbers of developers were sampled from teams of different sizes. The resulting sample represented teams of different sizes as well as developers with different backgrounds fairly well, as descriptive statistics in Chapter 5 show.

Nevertheless, Study 2 was an opt-in survey and thus participants were self-selected. Furthermore, the survey targeted active FOSS participants and focused on the current project the respondent spent most of their time working on. It may thus be possible that participants with more positive experiences of working in FOSS teams were more likely to participate, as those who may have experienced greater conflict levels could have left the team (Furumo, 2008). In fact, the mean surveyed levels of developers' intention to remain, identification with their

project and performance perception were all fairly high (between 3.8 and 4.0 out of 5), while conflict levels reported were fairly low (between 1.9 and 3.0 out of 5) suggesting this may indeed have been the case.

To partially address this limitation, the survey controls for elements such as tenure within the team, which did have a significant effect on levels of outcomes and emergent states reported. Nevertheless, this may somewhat limit the generalizability of present findings.

7.4.2 Time

Though Study 2 employs a cross-sectional research design to survey patterns in ongoing teams, Study 1 was designed to reinforce this limitation with respect to time, thereby providing context for the findings. Thus the interviews attempt to identify and trace conflict episodes from start to finish, as they have unfolded. Interviews in Study 1 required participants to recall subjective experiences of conflict episodes after the fact. To support this, participants were encouraged to share references to publicly available supporting materials (such as archival communication data) concerning these conflict episodes. This researcher further independently sought additional references related to these episodes, where this was possible, to support participant recollections with more objective accounts as well as to trace the development of the conflict episodes over time.

Taken together, the two studies provide some initial evidence for time effects of conflict in ongoing virtual teams, an area ripe for future work to explore in expanding and validating the present findings. In particular, now that the intragroup conflict taxonomy from traditional team research has been established to be relevant in FOSS, and updated for ongoing virtual teams, it would be worthwhile to observe the progress of the four kinds of conflict over time and trace more precisely their impact on team development and future structural changes.

7.4.3 Objective outcome and input measures

Furthermore, the present research did not employ objective group-level metrics of structural inputs, outcomes and emergent states, focusing instead on measuring individual perceptions of these variables. This is an intentional choice and tradeoff in the research design, as one of the present study's aims is to understand the interpretation of developers concerning conflict in their teams, as well as the way this impacts their intentions to remain in the team, sense of belonging with the team, and evaluation of the team's progress, all relatively subjective constructs. That said, it would be most relevant for future studies on conflict in ongoing virtual teams to examine more objective input and outcome measures connected with the entire corpus of team interactions, rather than just members who self-select to participate in a research study.

7.4.4 Multi-level effects

The present study highlights some interesting implications of multiple organizational levels on conflict manifestation (dyadic, intragroup and intergroup). For example, familiarity with different programming paradigms that emerges from contributing to multiple projects over the course of a developer's career has been shown to be both a potentially positive and negative input toward conflict. On one hand, bringing in experience from other projects and ways of doing things increases cognitive diversity and task conflict, resulting in potentially higher quality ideas. On the other hand, strong ideological roots in any one methodology, implementation, or framework may lead to unproductive and protracted debates between two or more equally good (or bad) options that may not be resolved without intervention.

As such, the influence of intergroup effects and the broader project context appear to be highly relevant to the experience of voluntary ongoing virtual teams such as FOSS projects. However, this is not an explicit part of the study design, and Study 2 does not explicitly take into account the impact of factors such as

subgroup formation, intergroup conflict, influences from other projects and the broader context the project belongs to. Furthermore, cross-level research design is an emerging trend in work on collocated teams and conflict (Korsgaard et al., 2008) and research on virtual teamwork and conflict may be able to draw from and contrast out findings with those of collocated teams.

For instance, there may be competing FOSS projects that work toward filling a similar function, such as media players, programming languages, or entire operating systems that spark intergroup conflicts. As the Node.JS example suggests, there may also be conflicts between companies supporting FOSS projects and the projects themselves. These sort of effects may result in intergroup conflict by encouraging stereotypical perceptions of the out-group and greater conflict between parties (Tajfel, 1978). Thus it would be interesting for future work on conflict both in FOSS, and more generally for ongoing virtual teams, to examine these sort of cross-level effects in more detail, possibly drawing on the framework used by Korsgaard et al. (2008) for collocated groups.

7.4.5 Conflict resolution

Finally, the present work does not take into account the effect of conflict resolution. Because this project focused on the emergence and manifestation of conflict, conflict resolution was outside of the intended scope of this research project. However, conflict resolution is another important team process that has a significant impact on the relationship between conflict types, team outcomes and emergent states. Specifically, research has identified 5 types of conflict resolution strategies: competitive, avoiding, accommodating, compromising and integrating (Blake & Mouton, 1964; Thomas & Schmidt, 1976). Studies have shown that similar to work on collocated teams, short-term virtual teams experienced higher performance when using a collaborative (or integrating) conflict management style (Montoya-Weiss et al., 2001; Paul, Samarah, Seetharaman, & Mykytyn, 2005). However, short term virtual teams were also found to ben-

efit from a competitive conflict resolution style, which may be a feature of their time orientation – under conditions of time pressure, and no expectation of future interaction with one another, teams may choose conflict resolution styles that lead to faster decisions even if they are perceived as less fair or distributive (Montoya-Weiss et al., 2001).

It would thus be interesting for future research to examine conflict resolution in the context of ongoing teams, to understand how expectations of future interaction affect conflict resolution choices in distributed settings, as well as trace how conflict resolution norms form and evolve. Are there specific team characteristics or behaviors that are more likely to encourage the development of collaborative conflict norms over time? For instance, Paul et al. (2005) highlight that conflict resolution style may be a product of the team's cultural norms, and find that short-term virtual teams that have a more collectivist orientation are more likely to employ collaborative conflict resolution strategies. At the same time, Martinez-Moreno, Zornoza, Orengo, and Thompson (2015) recently found that even when short-term virtual teams are given feedback about team processes, they are able improve their group's group conflict management strategies. There may be a connection between the conflicts a team experiences, and the subsequent evolution of conflict management strategies.

Taken together, addressing the above limitations may require the ability to trace conflict episodes over time objectively, while taking into account resolution mechanisms and multi-level effects. Free and Open Source Software development teams are fertile ground for this, as they often communicate through publicly available channels and carefully document all interactions for future use. Thus a recommended direction for future work is to make use of the publicly available archived FOSS project communication logs to carefully trace these more complex effects over time. Such a study can build on the foundations established in this work linking developer perceptions, conflict interpretations and their subsequent behavioral intentions, and connect this with rich objective

measures across time. It may also afford the ability to observe effects and their implications across different levels.

7.5 Conclusion

While virtual team research is moving forward in understanding the specific nature of intragroup processes in ongoing teams, there is as yet room to learn more. The present dissertation has taken the first steps to trace conflict emergence and evolution in ongoing virtual teams, and shown normative conflict to be one such critical transition process. Findings have highlighted that conflict can be interpreted in different ways by different group members, transform in a multitude of possible ways, and affect future team interactions, as well as more formal social and technical processes. In doing so the present work has expanded work on virtual team processes by drawing on work done in the unmediated organizational team space, as well as advances in computer-mediated-communication theory and social psychology. It is the hope of this researcher that the framework developed in this study would serve as a means for work on open collaborative systems and ongoing virtual teams to build more complex models and allow for comparative research.

References

About Debian. (n.d.).Retrieved from [2010-02-20]http://www.debian.org/intro/about

Acar, F. P. (2010). Analyzing the effects of diversity perceptions and shared leadership on emotional conflict: a dynamic approach. *The International Journal of Human Resource Management*, *21*(10), 1733–1753.

Ajzen, I. (2011). The theory of planned behaviour: Reactions and reflections. *Psychology & Health*, *26*(9), 1113–1127.

Ajzen, I., & Fishbein, M. (1977). Attitude-behavior relations: A theoretical analysis and review of empirical research. *Psychological Bulletin*, *84*(5), 888–918.

Amason, A. C. (1996). Distinguishing the Effects of Functional and Dysfunctional Confluct on Strategic Decision Making: Resolving a Paradox for Top Management Teams. *Academy of Management Journal*, *39*(1), 123–148.

Amason, A. C., & Sapienza, H. J. (1997). The Effects of Top Management Team Size and interaction Norms on Cognitive and Affective Conflict. *Journal of Management*, *23*(4), 495–516.

American Community Survey (Tech. Rep.). (2014). U.S. Census Bureau.Retrieved from https://www.census.gov/programs-surveys/acs/

Amrit, C., & Van Hillegersberg, J. (2010). Exploring the impact of socio-technical core-periphery structures in open source software development. *Journal of Information Technology*, *25*(2), 216–229.

Arazy, O., Nov, O., Patterson, R., & Yeo, L. (2011). Information Quality in Wikipedia: The Effects of Group Composition and Task Conflict. *Journal of Management Information Systems*, *27*(4), 71–98.

Arazy, O., Yeo, L., & Nov, O. (2013). Stay on the Wikipedia task: When task-related disagreements slip into personal and procedural conflicts. *Journal of the American Society for Information Science and Technology*, *64*(8), 1634–1648.

Arjona-Reina, L., Robles, G., & Dueñas, S. (2014). *The FLOSS 2013 Free/Libre/Open Source Survey* (Tech. Rep. No. 2).

Arrow, H., Poole, M. S., Henry, K. B., Wheelan, S., & Moreland, R. (2004). Time, Change, and Development: The Temporal Perspective on Groups. *Small Group Research*, *35*(1), 73–105.

Ashforth, B. E., & Mael, F. (1989, jan). Social Identity Theory and the Organization. *The Academy of Management Review*, *14*(1), 20.

Avolio, B. J., Bass, B. M., & Jung, D. I. (1999, dec). Re-examining the components of transformational and transactional leadership using the Multifactor Leadership. *Journal of Occupational and Organizational Psychology*, *72*(4), 441–462.

Avolio, B. J., Kahai, S., & Dodge, G. E. (2000). E-leadership: Implications for theory, research and practice. *The Leadership Quarterly*, *11*(4), 615–668.

Avolio, B. J., & Kahai, S. S. (2003). Adding the "E" to E-leadership: Hot it may impact your leadership. *Organizational Dynamics*, *31*(4), 325–338.

Axelrod, R., & Hamilton, W. D. (1981). The Evolution of Cooperation. *Science*, *211*(March), 1390–1396.

Ayoko, O. B., Konrad, A. M., & Boyle, M. V. (2012). Online work: Managing conflict and emotions for performance in virtual teams. *European Management Journal*, *30*(2), 156–174.

Bagozzi, R. P., & Dholakia, U. M. (2005). Open Source Software User Communities: A Study of Participation in Linux User Groups. *Management Science*, *52*(7), 1099 – 1115.

Bass, B. M. (1985). *Leadership and performance beyond expectations.* New York: Free Press.

Bass, B. M. (1990, dec). From transactional to transformational leadership: Learning to share the vision. *Organizational Dynamics*, *18*(3), 19–31.

Baytiyeh, H., & Pfaffman, J. (2010). Open source software: A community of altruists. *Computers in Human Behavior*, *26*(6), 1345–1354.

Behfar, K. J., Mannix, E. A., Peterson, R. S., & Trochim, W. M. (2011). Conflict in Small Groups: The Meaning and Consequences of Process Conflict. *Small Group Research*, *42*(2), 127–176.

Behfar, K. J., Peterson, R. S., Mannix, E. A., & Trochim, W. M. K. (2008). The critical role of conflict resolution in teams: a close look at the links between conflict type, conflict management strategies, and team outcomes. *The Journal of applied psychology*, *93*(1), 170–188.

Beiser, F. C. (2005). *Hegel*. London: Routledge.

Bell, B. S., & Kozlowski, S. W. J. (2002). A Typology of Virtual Teams: Implications for Effective Leadership. *Group and Organization Management*, *27*(1), 14–49.

Bezroukov, N. (1999). Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism). *First Monday*, *4*, 1–16.

Bishop, J. (2000). Support, commitment, and employee outcomes in a team environment. *Journal of Management*, *26*(6), 1113–1132.

Blake, R. R., & Mouton, J. (1964). *The managerial grid: The key to leadership excellence*. Houson: Gulf Publishing Company.

Blake, R. R., & Mouton, J. S. (1984). Overcoming Group Warfare. *Harvard Buisness Review*, *Nov-Dec*(December 1964), 98–108.

Bolici, F., Howison, J., Crowston, K., Osterlund, C., Crowston, K., & Bolici, F. (2009). Coordination without discussion? Socio-technical congruence and Stigmergy in Free and Open Source Software projects. In *2nd international workshop on socio-technical congruence, icse* (pp. 1 – 9). Vancouver, BC.

Boone, C., & Hendriks, W. (2008). Top Management Team Diversity and Firm Performance: Moderators of Functional-Background and Locus-of-Control Diversity. *Management Science*, *55*(2), 165–180.

Bosch-sijtsema, P. (2007). The Impact of Individual Expectations and Expectation Conflicts on Virtual Teams. *Group & Organization Management*, *32*(3), 358–389.

Breckenridge, J. P., Jones, D., Elliott, I., & Nicol, M. (2012). Choosing a Methodological Path: Reflections on the Constructivist Turn. *Grounded Theory Review*, *11*(1).

Brett, J. M. (1984). Managing organizational conflict. *Professional Psychology: Research and Practice*, *15*(5), 664–678.

Campbell, J., Fletcher, G., & Greenhill, A. (2009). Conflict and identity shape shifting in an online financial community. *Information Systems Journal*, *19*(5), 461–478.

Campion, M., Papper, E., & Medsker, G. (1996). Relations Between Work Team Characteristics and Effectiveness: a Replication and Extension. *Personnel Psychology*, *49*(2), 429–452.

Carless, S. A., Wearing, A. J., & Mann, L. (2000). A short measure of transformational leadership. *Journal of Business and Psychology*, *14*(3), 389–405.

Carson, J. B., Tesluk, P. E., & Marrone, J. a. (2007). Shared Leadership in Teams: an Investigation of Antecedent Conditions and Performance. *Academy of Management Journal*, *50*(5), 1217–1234.

Casaló, L. V., Cisneros, J., Flavián, C., & Guinalíu, M. (2009). Determinants of success in open source software networks. *Industrial Management & Data Systems*, *109*(4), 532–549.

Cass, S. (2015). *The 2015 Top Ten Programming Languages*.Retrieved from [2015-09-01]`http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages`

Chan, D. (1998). Functional relations among constructs in the same content

domain at different levels of analysis: A typology of composition models. *Journal of Applied Psychology*, *83*(2), 234–246.

Charmaz, K. (2003). Grounded theory. In J. Smith (Ed.), *Qualitative psychology: A practical guide to research methods* (2nd ed., pp. 81–110). London: Sage.

Chen, G., Sharma, P. N., Edinger, S. K., Shapiro, D. L., & Farh, J.-L. (2011). Motivating and demotivating forces in teams: Cross-level influences of empowering leadership and relationship conflict. *Journal of Applied Psychology*, *96*(3), 541–557.

Chidambaram, L. (1996). Relational Development in Computer-Supported Groups. *MIS Quarterly*, *20*(June), 143–166.

Chidambaram, L., Bostrom, R. P., & Wynne, B. E. (1991). A longitudinal study of the impact of group decision support systems on group development. *Journal of Management Information Systems*, *7*(3), 7–25.

Choi, S., & Schnurr, S. (2013). Exploring distributed leadership: Solving disagreements and negotiating consensus in a 'leaderless' team. *Discourse Studies*, *16*(1), 3–24.

Chou, S., & He, M. Y. (2010). The factors that affect the performance of open source software development: The perspective of social capital and expertise integration. *Information Systems Journal*, *21*(2), 195–219.

Chudoba, K. M., Wynn, E., Lu, M., & Watson-Manheim, M. B. (2005). How virtual are we? Measuring virtuality and understanding its impact in a global organization. *Information Systems Journal*, *15*(4), 279–306.

Cialdini, R. B., Kallgren, C. A., & Reno, R. R. (1991). A Focus Theory of Normative Conduct: A Theoretical Refinement and Reevaluation of the Role of Norms in Human Behavior. In *Advances in experimental social psychology* (Vol. 24, pp. 201–234). London: Academic Press.

Cohen, J. (1992). A power primer. *Psychological bulletin*, *112*(1), 155–159.

Colazo, J. A., & Fang, Y. (2010). Following the Sun : Temporal Dispersion and Performance in Open Source Software Project Teams. *Journal of the Association for Information Systems*, *11*(Special Issue), 684–707.

Cole, M. S., & Bruch, H. (2006). Organizational identity strength, identification, and commitment and their relationships to turnover intention: Does organizational hierarchy matter? *Journal of Organizational Behavior*, *27*(5), 585–605.

Collier, B., & Bear, J. (2012). Conflict, Confidence, or Criticism : An Empirical Examination of the Gender Gap in Wikipedia. In *Cscw'12: Proceedings of the acm 2012 conference on computer supported collaborative work* (pp. 383–392). Seattle, Washington, USA.

Conger, J. A., & Kanungo, R. N. (1988). The Empowerment Process: Integrating Theory and Practice. *Academy of Management Review*, *13*(3), 471–482.

Coser, L. (1957). Social Conflict and the Theory of Social Change. *The British Journal of Sociology*, *8*(3), 197–207.

Cramton, C. D. (2001). The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. *Organization Science*, *12*(3), 346–371.

Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. In *Proceedings of the 24th international conference on information systems* (pp. 1–28).

Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, *18*(4), 65–85.

Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process Improvement and Practice*, *11*(2), 123–148.

Crowston, K., Howison, J., Masango, C., & Eseryel, U. Y. (2007). The role of face-to-face meetings in technology-supported self-organizing distributed teams. *IEEE Transactions on Professional Communication*, *50*(3), 185–203.

Crowston, K., Kangning Wei, Qing Li, & Howison, J. (2006). Core and Periphery in Free/Libre and Open Source Software Team Communications. In *Proceedings of the 39th annual hawaii international conference on system sciences (hicss'06)* (pp. 1–7). IEEE.

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre opensource software development. *ACM Computing Surveys*, *44*(2), 1–35.

Culnan, M. J., & Markus, M. L. (1987). Information Technologies. In F. M. Jablin, L. L. Putnam, H. Roberts, Karlene, & L. W. Porter (Eds.), *Handbook of organizational communication: An iinterdisciplinary perspective* (pp. 420–433). Thousand Oaks, CA: Sage Publications, Inc.

Curşeu, P. L. (2006). Emergent states in virtual teams: A complex adaptive systems perspective. *Journal of Information Technology*, *21*(4), 249–261.

Dabbish, L., Farzan, R., Kraut, R., & Postmes, T. (2012). Fresh faces in the crowd. In *Proceedings of the acm 2012 conference on computer supported cooperative work - cscw '12* (p. 245). New York.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub. In *Proceedings of the acm 2012 conference on computer supported cooperative work - cscw '12* (p. 1277). New York.

Daft, R. L., & Lengel, R. H. (1984). Information Richness: A New Approach

to Managerial Behavior and Organization Design. In L. L. Cummings & B. Staw (Eds.), *Research in organizational behavior* (Vol. 6, pp. 191–233). Homewood, IL: JAI Press.

Dayan, M. (2010). Managerial trust and NPD team performance: team commitment and longevity as mediators. *Journal of Business & Industrial Marketing*, *25*(2), 94–105.

De Dreu, C. K. W. (2006). When Too Little or Too Much Hurts: Evidence for a Curvilinear Relationship Between Task Conflict and Innovation in Teams. *Journal of Management*, *32*(1), 83–107.

De Dreu, C. K. W., & Weingart, L. R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *The Journal of Applied Psychology*, *88*(4), 741–749.

De Dreu, C. K. W., & West, M. a. (2001). Minority dissent and team innovation: the importance of participation in decision making. *The Journal of applied psychology*, *86*(6), 1191–1201.

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean Model of Information Systems Success : A Ten-Year Update. *Journal of Management Information Systems*, *19*(4), 9–30.

DeSanctis, G., & Monge, P. (2006). Communication Processes for Virtual Organizations. *Journal of Computer-Mediated Communication*, *3*(4), 0.

DeSanctis, G., & Poole, M. S. (1994, may). Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. *Organization Science*, *5*(2), 121–147.

Deutsch, M. (1949). A Theory of Co-operation and Competition. *Human Relations*, *2*(2), 129–152.

Dewey, J. (1922). *Human Nature and Conduct: An introduction to social psychology*. New York: Henry Holt and Company.

de Wit, F. R. C., Greer, L. L., & Jehn, K. A. (2012). The paradox of intragroup conflict: A meta-analysis. *Journal of Applied Psychology*, *97*(2), 360–390.

Dionne, S. D., Yammarino, F. J., Atwater, L. E., & Spangler, W. D. (2004, apr). Transformational leadership and team performance. *Journal of Organizational Change Management*, *17*(2), 177–193.

Dodoiu, G. (2015). Intentions for cooperative conflict resolution in groups. *Team Performance Management: An International Journal*, *21*(5/6), 259–273.

Duchenaut, N. (2005). Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work (CSCW)*, *14*(4), 323–368.

Durlak, J. A. (2009, oct). How to Select, Calculate, and Interpret Effect Sizes. *Journal of Pediatric Psychology*, *34*(9), 917–928.

Ehmke, C. A. (2014). *Contributor Covenant.*Retrieved from [15-10-2015]`http://contributor-covenant.org`

Ehmke, C. A. (2016). *Code of Conduct.*Retrieved from [15-2-2016]`https://bugs.ruby-lang.org/issues/12004`

Elliott, M. S., & Scacchi, W. (1993). Communicating and Mitigating Conflict in Open Source Software Development Projects. *Projects and Profits*, *10*(1), 25–41.

Elliott, M. S., & Scacchi, W. (2003). Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration. In *Proceedings of the 4th international conference on supporting group work* (pp. 21–30). Sanibel Island, Florida.

Emerson, R. M. (1962). Power-Dependence Relations. *American Sociological Review*, *27*(1), 31–40.

Erenkrantz, J. R. (2003). Release Management Within Open Source Projects. In *Proceedings of the 3rd workshop on open source software* (pp. 51–55).

Falk, A., & Fischbacher, U. (2006). A theory of reciprocity. *Games and Economic Behavior*, *54*(2), 293–315.

Fang, Y., & Neufeld, D. (2009). Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, *25*(4), 9–50.

Farh, J.-L., Lee, C., & Farh, C. I. C. (2010, nov). Task conflict and team creativity: A question of how much and when. *Journal of Applied Psychology*, *95*(6), 1173–1180.

Fayol, H. (1949). *General And Industrial Management*. London: Pitman.

Feldman, D. C. (1984). The Development and Enforcement of Group Norms. *Academy of Management Review*, *9*(1), 47–53.

Feller, J., & Fitzgerald, B. (2002). *Understanding Open Source Software Development*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Fielding, R. T. (1999). Shared Leadership. *Communications of the ACM*, *42*(4), 42–43.

Filippova, A., & Cho, H. (2015). Mudslinging and Manners: Unpacking conflict in Free and Open Source Software. In *Proceedings of the 18th acm conference on computer supported cooperative work & social computing - cscw '15* (pp. 1393–1403). Vancouver, BC: ACM Press.

Filippova, A., & Cho, H. (2016). The Effects and Antecedents of Conflict in Free and Open Source Software Development. In *Proceedings of the*

19th acm conference on computer supported cooperative work & social computing - cscw '16. San Francisco, CA: ACM Press.

Fink, C. F. (1968). Some conceptual difficulties in the theory of social conflict. *Journal of Conflict Resolution*, *12*(4), 412–460.

Fiol, C. M., & O'Connor, E. J. (2005). Identification in Face-to-Face, Hybrid, and Pure Virtual Teams: Untangling the Contradictions. *Organization Science*, *16*(1), 19–32.

Franco, V., Piirto, R., Hu, H. Y., Lewenstein, B. V., Underwood, R., & Vidal, N. K. (1995). Anatomy of a flame: conflict and community building on the internet. *IEEE Technology and Society Magazine*, *14*(2), 12–21.

Friedman, R. A., & Currall, S. C. (2003). Conflict Escalation: Dispute Exacerbating Elements of E-mail Communication. *Human Relations*, *56*(11), 1325–1347.

Fuchs, C. (2008). *Internet and Society*. London: Routledge.

Furumo, K. (2008). The Impact of Conflict and Conflict Management Style on Deadbeats and Deserters in Virtual Teams. *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, 1–10.

Gersick, C. (1988). Time And Transition in Work Teams: Towards a New Model Of Group Development. *Academy of Management Journal*, *31*(1), 9–41.

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). *Free/Libre and open source software: Survey and study-Deliverable D18: Final report-Part IV: Survey of developers* (Tech. Rep.). International Institute of Infonomics University of Maastricht, The Netherlands.

Ghosh, T., Yates, J., & Orlikowski, W. (2004). Using Communication Norms for Coordination: Evidence from a Distributed Team. *ICIS 2004 Proceedings*, 115–127.

Gibb, C. A. (1954). Leadership. In G. Lindzey (Ed.), *Handbook of social psychology* (pp. 877–917). Reading, MA: Addison-Wesley.

Giddens, A. (1986). *The Constitution of Society: Outline of the Theory of Structuration*. University of California Press.Retrieved from `https://books.google.com/books?id=cV8xnSIa0-IC`

Gilson, L. L., Maynard, M. T., Jones Young, N. C., Vartiainen, M., & Hakonen, M. (2014). Virtual Teams Research: 10 Years, 10 Themes, and 10 Opportunities. *Journal of Management*, *41*(5), 1313–1337.

Giuri, P., Rullani, F., & Torrisi, S. (2008). Explaining leadership in virtual teams: The case of open source software. *Information Economics and Policy*, *20*(4), 305–315.

Glaser, B. G. (1978). *Theoretical sensitivity: advances in the methodology of grounded theory*. Mill Valley, CA: Sociology Press.

Glaser, B. G., & Strauss, A. (1967). *The discovery of grounded theory. 1967*. London: Weidenfield & Nicolson.

Glasl, F. (1982). The process of conflict escalation and roles of third parties. In G. B. J. Bomers & R. Peterson (Eds.), *Conflict management and industrial relations* (pp. 119– 140). Boston: Kluwer-Nijhoff.

Gouldner, A. W. (1960). The Norm of Reciprocity: a Preliminary Statement. *American Sociological Review*, *25*(2), 161.

Gousios, G. (2013). The GHTorrent dataset and tool suite. In *Proceedings of the 10th working conference on mining software repositories* (pp. 233–236). Piscataway, NJ, USA: MSR '13.

Greene, S. (2014). Understanding Party Identification: A Social Identity Approach. *Political Psychology*, *20*(2), 393–403.

Greer, L. L., Jehn, K. A., & Mannix, E. A. (2008). Conflict Transformation: A Longitudinal Investigation of the Relationships Between Different Types of Intragroup Conflict and the Moderating Role of Conflict Resolution. *Small Group Research*, *39*(3), 278–302.

Griffith, T. L., Sawyer, J. E., & Neale, M. A. (2003). Virtualness and Knowledge in Teams: Managing the Love Triangle of Organizations, Individuals, and Information Technology. *MIS Quarterly*, *27*(2), 265–287.

Guetzkow, H., & Gyr, J. (1954). An Analysis of Conflict in Decision-Making Groups. *Human Relations*, *7*(3), 367–382.

Gully, S. M., Incalcaterra, K. a., Joshi, A., & Beauien, J. M. (2002). A meta-analysis of team-efficacy, potency, and performance: interdependence and level of analysis as moderators of observed relationships. *The Journal of applied psychology*, *87*(5), 819–832.

Guzzo, R. A., Yost, P. R., Campbell, R. J., & Shea, G. P. (1993). Potency in groups: Articulating a construct. *British Journal of Social Psychology*, *32*(1), 87–106.

Hackathorn, R. D., & Keen, P. G. W. (1981). Organizational Strategies for Personal Computing in Decision Support Systems. *MIS Quarterly*, *5*(3), 21–27.

Hackman, J. R. (1987). The design of work teams. In J. Lorsch (Ed.), *Handbook of organizational behavior* (pp. 315–342). Englewood Cliffs, NJ: Prentice-Hall.

Hall, E. T. (1976). *Beyond culture*. Garden City: Anchor Press.

Hambley, L. A., O'Neill, T. A., & Kline, T. J. (2007). Virtual team leadership: The effects of leadership style and communication medium on team

interaction styles and outcomes. *Organizational Behavior and Human Decision Processes*, *103*(1), 1–20.

Hamner, H. H. (1987). A functional taxonomy of normative conflict. *Law and Philosophy*, *6*, 227–247.

Han, G. H., & Harms, P. (2010). Team identification, trust and conflict: a mediation model. *International Journal of Conflict Management*, *21*(1), 20–43.

Hars, A., & Ou, S. O. S. (2001). Working for free? Motivations of participating in open source projects. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*.

Heath, H., & Cowley, S. (2004). Developing a grounded theory approach: A comparison of Glaser and Strauss. *International Journal of Nursing Studies*, *41*(2), 141–150.

Hemetsberger, A., & Reinhardt, C. (2004). Sharing and creating knowledge in open-source communities: the case of KDE. In *Fifth european conference on organizational knowledge, learning, and capabilities* (pp. 1–8). Innsbruck, Austria.

Hendrickson, M., Magoulas, R., & O'Reilly, T. (2012). *Economic Impact of Open Source on Small Business: A Case Study.* O'Reilly Media.

Hill, W. F., & Gruner, L. (1973). A Study of Development in Open and Closed Groups. *Small Group Research*, *4*(3), 355–381.

Hinds, P. J., & Bailey, D. E. (2003). Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams. *Organization Science*, *14*(6), 615–632.

Hinds, P. J., & McGrath, C. (2006). Structures that work. In *Proceedings of the 2006 20th anniversary conference on computer supported cooperative work - cscw '06* (p. 343). New York, New York, USA: ACM Press.

Hinds, P. J., & Mortensen, M. (2005). Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science*, *16*(3), 290–307.

Hirokawa, R. Y., & Rost, K. M. (1992). Effective group decision making in organizations. *Management Communication Quarterly*, *5*(3), 267.

Hobbes, T. (1651). *Leviathan or the Matter, Forme, & Power of a Common-Wealth Ecclesiastical and Civil.* Printed for Andrew Crooke, at the Green Dragon in St. Paul's Churchyard.

Hoch, J. E., & Kozlowski, S. W. J. (2014). Leading virtual teams: hierarchical leadership, structural supports, and shared team leadership. *The Journal of applied psychology*, *99*(3), 390–403.

Howison, J. (2009). *Alone Together: A socio-technical theory of motivation,*

coordination and collaboration technologies in organizing for free and open source software development (Unpublished doctoral dissertation). Syracuse University.

Huang, J. (2010). Unbundling task conflict and relationship conflict. *International Journal of Conflict Management*, *21*(3), 334–355.

IETF. (2009). *Request for comment.*Retrieved from [2009-05-20]`https://www.ietf.org/rfc.html`

Ilgen, D. R., Hollenbeck, J. R., Johnson, M., & Jundt, D. (2005). Teams in Organizations: From Input-Process-Output Models to IMOI Models. *Annual Review of Psychology*, *56*(1), 517–543.

J, A. (2016). *Comment on "Code of Conduct".*Retrieved from [15-2-2016]`https://bugs.ruby-lang.org/issues/12004{#}note-16`

Jehn, K. A. (1994). Enhancing Effectiveness: An Investigation of Advantages And Disadvantages of ValueBased Intragroup Conflict. *International Journal of Conflict Management*, *5*(3), 223–238.

Jehn, K. A. (1995). A multimethod examination of the benefits and detriments of intragroup conflict. *Administrative Science Quarterly*, *40*(2), 256–282.

Jehn, K. A. (1997). A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups. *Administrative Science Quarterly*, *42*(3), 530–557.

Jehn, K. A., & Bendersky, C. (2003). Intragroup conflict in organizations: a contingency perspective on the conflict-outcome relationship. *Research in Organizational Behavior*, *25*, 187–242.

Jehn, K. A., Greer, L., Levine, S., & Szulanski, G. (2008). The effects of conflict types, dimensions, and emergent states on group outcomes. *Group Decision and Negotiation*, *17*(6), 465–495.

Jehn, K. A., & Mannix, E. A. (2001). The dynamic nature of conflict: A Longitudinal Study of Intragroup Conflict and Group Performance. *Academy of Management Journal*, *44*(2), 238–251.

Jehn, K. A., Northcraft, G. B., & Neale, M. A. (1999). Why Differences Make a Difference: A Field Study of Diversity, Conflict, and Performance in Workgroups. *Administrative Science Quarterly*, *44*(4), 741–763.

Jehn, K. A., Rispens, S., Jonsen, K., & Greer, L. (2013). Conflict contagion: a temporal perspective on the development of conflict within teams. *International Journal of Conflict Management*, *24*(4), 352–373.

Jensen, C., & Scacchi, W. (2005). Collaboration, Leadership, Control, and Conflict Negotiation and the Netbeans.org Open Source Software Development Community. In *Proceedings of the 38th annual hawaii interna-*

tional conference on system sciences (pp. 1–10).

Jones, M., & Karsten, H. (2003). Review: Structuration theory and information systems research. *Research Papers in Management Studies*, *11*, 85.Retrieved from `http://papers.ssrn.com/sol3/papers.cfm?abstract{_}id=424062`

Jong, R. D., Schalk, R., & Curseu, P. L. (2008). Virtual communicating, conflicts and performance in teams. *Team Performance Management*, *14*(7/8), 364–380.

Judge, T. A., & Piccolo, R. F. (2004). Transformational and transactional leadership: a meta-analytic test of their relative validity. *The Journal of applied psychology*, *89*(5), 755–768.

Kamp, P.-H. (2003). *The Bikeshed email.*Retrieved from [2015-09-01]`http://phk.freebsd.dk/sagas/bikeshed.html`

Kankanhalli, A., Tan, B. C., & Wei, K.-K. (2007). Conflict and Performance in Global Virtual Teams. *Journal of Management Information Systems*, *23*(3), 237–274.

Katzenbach, J. R., & Smith, D. K. (1993). The Discipline of Teams. *Harvard Buisness Review*(March-April), 111–120.

Ke, W., & Zhang, P. (2011, may). Effects of empowerment on performance in open-source software projects. *IEEE Transactions on Engineering Management*, *58*(2), 334–346.

Kelty, C. M. (2008). *Two Bits: The Cultural Significance of Free Software*. Duke University Press.

Kevin, C., Annabi, H., Howison, J., Masango, C., & Crowston, K. (2004). Effective Work Practices for Software Engineering: Free/Libre Open Source Software Development. *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*, 18–26.

Kiesler, S., Zubrow, D., Moses, A. M., & Geller, V. (1985, mar). Affect in Computer-Meditated Communication: An Experiment in Synchronous Terminal-to-Terminal Discussion. *Human–Computer Interaction*, *1*(1), 77–104.

Kilmann, R. H., & Thomas, K. W. (1977). Developing a Forced-Choice Measure of Conflict-Handling Behavior: The "Mode" Instrument. *Educational and Psychological Measurement*, *37*(2), 309–325.

Kirkman, B. L. (2005). The Dimensions and Antecedents of Team Virtuality. *Journal of Management*, *31*(5), 700–718.

Kittur, A., Chi, E. H., & Suh, B. (2009). What's in Wikipedia? In *Proceedings of the 27th international conference on human factors in computing systems - chi 09* (p. 1509). New York, New York, USA: ACM Press.

Kittur, A., & Kraut, R. E. (2010). Beyond Wikipedia: Coordination and Conflict in Online Production Groups. *Communication*, *34*(2), 215–224.

Kittur, A., Suh, B., Pendleton, B. A., & Chi, E. H. (2007). He says, she says: conflict and coordination in Wikipedia. *ACM Conference on Human Factors in Computing Systems*, 453 – 462.

Korsgaard, M. A., Soyoung Jeong, S., Mahony, D. M., & Pitariu, A. H. (2008). A Multilevel View of Intragroup Conflict. *Journal of Management*, *34*(6), 1222–1252.

Kotlyar, I., & Karakowsky, L. (2006). Leading Conflict? Linkages Between Leader Behaviors and Group Conflict. *Small Group Research*, *37*(4), 377–403.

Kozlowski, S. W. J., & Bell, B. S. (2001). Work Groups and Teams in Organizations Work Groups and Teams in Organizations. *ILR Collection*(2003), 1–70.

Krishnamurthy, S. (2005). Cave or Community? An empirical Examination of 100 Mature Open Source Projects. *First Monday*(Special Issue #2: Open Source).

Kuhn, T., & Poole, M. S. (2000). Do conflict management styles affect group decision making? *Human Communication Research*, *26*(4), 558–590.

Landau, J., & Hammer, T. H. (1986). Clerical Employees ' Perceptions of Intraorganizational Career Opportunities. *The Academy of Management Journal*, *29*(2), 385–404.

Langfred, C. W. (2007). The downside of self-management: A longitudinal study of the effects of conflict on trust, autonomy, and task interdependence in self-managing teams. *Academy of Management Journal*, *50*(4), 885–900.

Lau, D. C., & Murnighan, J. K. (2005). Interactions within groups and subgroups: The effects of demographic faultlines. *Academy of Management Journal*, *48*(4), 645–659.

Lea, M., & Spears, R. (1991). Computer-mediated communication, de-individuation and group decision-making. *Int. J. Man-Machine Studies*, *34*, 283–301.

Lee, C., Tinsley, H., & Bobko, P. (2002). An investigation of the antecedents and consequences of group-level confidence. *Journal of Applied Social Psychology*, *32*(8), 1628–1652.

Lee, S. T., Kim, H., & Gupta, S. (2009). Measuring open source software success. *Omega*, *37*(2), 426–438.

LePine, J. A., Picollo, R. F., Jackson, C. L., Matheiu, J. E., & Saul, J. R. (2008, jun). A Meta-analysis of Teamwork Processes: Tests of a Multidimen-

sional Model and Relationships with Team Effectiveness Criteria. *Personnel Psychology*, *61*(2), 273–307.

Li, Y., Tan, C. H., & Teo, H. H. (2012). Leadership characteristics and developers' motivation in open source software development. *Information and Management*, *49*(5), 257–267.

Lira, E. M., Ripoll, P., Peiró, J. M., & Orengo, V. (2008). How do different types of intragroup conflict affect group potency in virtual compared with face-to-face teams? A longitudinal study †. *Behaviour & Information Technology*, *27*(2), 107–114.

Little, T. D., Bovaird, J. A., & Widaman, K. F. (2006). On the Merits of Orthogonalizing Powered and Product Terms: Implications for Modeling Interactions Among Latent Variables. *Structural Equation Modeling*, *13*(4), 497–519.

Long, Y., & Siau, K. (2007). Social network structures in open source software development teams. *Journal of Database Management*, *18*(2), 25–40.

Lowrey, W. (2003). Normative Conflict in the Newsroom: The Case of Digital Photo Manipulation. *Journal of Mass Media Ethics*, *18*(2), 123–142.

Madey, G., Madey, G., Freeh, V., & Freeh, V. (2002). the Open Source Software Development Phenomenon: an Analysis Based on Social Network Theory. *Eighth Americas Conference on Information Systems*, 1806–1813.

Mannix, E., & Jehn, K. A. (2003). Let's Norm and Storm, But Not Right Now: Integrating Models of Group Development and Performance. *Research on Managing Groups and Teams*, *6*, 11–37.

Marks, M., Mathieu, J. E., & Zaccaro, S. J. (2001). A temporally based framework and taxonomy of team processes. *Academy of Management Review*, *26*(3), 356–376.

Marson, I. (2004). *Singapore Defence switches to OpenOffice.org*.Retrieved from [2008-02-20]`http://www.zdnet.com/singapore-defence-switches-to-openoffice-org-1139163830/`

Martinez-Moreno, E., González-Navarro, P., Zornoza, A., & Ripoll, P. (2009). Relationship, task and process conflicts on team performance: The moderating role of communication media. *International Journal of Conflict Management*, *20*(3), 251–268.

Martinez-Moreno, E., Zornoza, A., González-Navarro, P., & Thompson, L. F. (2012). Investigating face-to-face and virtual teamwork over time: When does early task conflict trigger relationship conflict? *Group Dynamics: Theory, Research, and Practice*, *16*(3), 159–171.

Martinez-Moreno, E., Zornoza, A., Orengo, V., & Thompson, L. F. (2015). The Effects of Team Self-Guided Training on Conflict Management in Virtual

Teams. *Group Decision and Negotiation*, *24*(5), 905–923.

Martins, L. L., Gilson, L. L., Maynard, M. T., & Martins, L. L. (2004). Virtual Teams: What Do We Know and Where Do We Go From Here? *Journal of Management*, *30*(6), 805–835.

Martins, L. L., & Schilpzand, M. C. (2011). *Global Virtual Teams: Key Developments, Research Gaps, and Future Directions* (Vol. 30). Emerald Group Publishing Ltd.

Matei, S. A., & Dobrescu, C. (2010, dec). Wikipedia's "Neutral Point of View": Settling Conflict through Ambiguity. *The Information Society*, *27*(1), 40–51.

Mathieu, J. E., & Rapp, T. L. (2009). Laying the foundation for successful team performance trajectories: The roles of team charters and performance strategies. *Journal of Applied Psychology*, *94*(1), 90–103.

Maynard, M. T., Mathieu, J. E., Rapp, T. L., & Gilson, L. L. (2012). Something(s) old and something(s) new: Modeling drivers of global virtual team effectiveness. *Journal of Organizational Behavior*, *33*(3), 342–365.

Mayo, E. (1933). *The human problems of an industrial civilization*. New York: McMillan Company.

McDonald, R. I., Fielding, K. S., & Louis, W. R. (2012). Energizing and De-Motivating Effects of Norm-Conflict. *Personality and Social Psychology Bulletin*, *39*(1), 52–72.

McDonald, R. P., & Ho, M.-H. R. (2002). Principles and practice in reporting structural equation analyses. *Psychological methods*, *7*(1), 64–82.

McDonough, E. F., Kahn, K. B., & Barczak, G. (2001). An investigation of the use of global, virtual, and colocated new product development teams. *Journal of Product Innovation Management*, *18*(2), 110–120.

McGrath, J. E. (1984). *Groups: Interaction and perfomance*. Englewood Cliffs, NJ: Prentice-Hall.

Michaels, C. E., & Spector, P. E. (1982). Causes of Employee Turnover : A Test of the Mobley, Griffeth, Hand, and Meglino Model. *Journal of Applied Psychology*, *67*(1), 53–59.

Midha, V., & Palvia, P. (2012). Factors affecting the success of Open Source Software. *Journal of Systems and Software*, *85*(4), 895–905.

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Softw. Eng. Meth.*, *11*(3), 309–346.

Montoya-Weiss, M. M., Massey, a. P., & Song, M. (2001). Getting It Together: Temporal Coordination and Conflict Management in Global Vir-

tual Teams. *Academy of Management Journal*, *44*(6), 1251–1262.

Moon, J. Y., & Sproull, L. (2002). Essence of distributed work: The case of the Linux kernel. *First Monday*, *10*, 381–404.

Mooney, A. C., Holahan, P. J., & Amason, A. C. (2007). Don't take it personally: Exploring cognitive conflict as a mediator of affective conflict. *Journal of Management Studies*, *44*(5), 733–758.

Moreland, R., & Levine, M. (1982). Socialization in small groups: Tem- poral changes in individual-group relations. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (pp. 137–192). New York: Academic Press.

Mortensen, M., & Hinds, P. J. (2001). Conflict and Shared Identity in Geographically Distributed Teams. *The International Journal of Conflict Management*, *12*(3), 212–238.

Mowday, R. T., Steers, R. M., & Porter, L. W. (1979). The measurement of organizational commitment. *Journal of Vocational Behavior*, *14*(2), 224–247.

Nelson, R. E. (1989). The Strength of Strong Ties: Social Networks and Intergroup Conflict in Organizations. *Academy of Management Journal*, *32*(2), 377–401.

Nemeth, C. J., & Goncalo, J. A. (2011). Rogues and Heroes: Finding value in dissent. In *Rebels in groups: Dissent, deviance, difference and defiance* (pp. 17–35). Oxford, UK: Blackwell Publishing.

Netcraft. (2015). *August 2015 Web Server Survey.*Retrieved from [2015-09-01]`http://news.netcraft.com/archives/2015/08/13/august-2015-web-server-survey.html`

Nov, O. (2007). What motivates Wikipedians? *Communications of the ACM*, *50*(11), 60–64.

Nunnally, J. (1978). *Psychometric theory*. New York: McGraw-Hill.

O'Leary, M. B., & Mortensen, M. (2010). Go (Con)figure: Subgroups, Imbalance, and Isolates in Geographically Dispersed Teams. *Organization Science*, *21*(1), 115–131.

Olson, B. J., Parayitam, S., & Yongjian Bao. (2007). Strategic Decision Making: The Effects of Cognitive Diversity, Conflict, and Trust on Decision Outcomes. *Journal of Management*, *33*(2), 196–222.

O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal*, *50*(5), 1079–1106.

O'Neill, T. A., Allen, N. J., & Hastings, S. E. (2013). Examining the "Pros" and "Cons" of Team Con flict: A Team-Level Meta-Analysis of Task,

Relationship, and Process Conflict. *Human Performance*, *26*(3), 236–260.

Opp, K.-D. (1982). The evolutionary emergence of norms. *British Journal of Social Psychology*, *21*(2), 139–149.

Oreg, S., & Nov, O. (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in Human Behavior*, *24*(5), 2055–2073.

Oren, N., Luck, M., Miles, S., & Norman, T. J. (2010). An Argumentation Inspired Heuristic for Resolving Normative Conflict. In *Proceedings of the fifth international workshop on coordination, organization, institutions and norms in agent systems* (pp. 1–15).

Ortiz De Guinea, A., Webster, J., & Staples, D. S. (2012). A meta-analysis of the consequences of virtualness on team functioning. *Information and Management*, *49*(6), 301–308.

Overview of the GNU System. (n.d.).Retrieved from `http://www.gnu.org/gnu/gnu-history.html`

Packer, D. J. (2007). On being both with us and against us: a normative conflict model of dissent in social groups. *Personality and Social Psychology Review*, *12*(1), 50–72.

Packer, D. J., & Chasteen, A. L. (2010). Loyal Deviance: Testing the Normative Conflict Model of Dissent in Social Groups. *Personality and Social Psychology Bulletin*, *36*(1), 5–18.

Park, H., & Antonioni, D. (2007). Personality, reciprocity, and strength of conflict resolution strategy. *Journal of Research in Personality*, *41*(1), 110–125.

Parkinson, C. N. (1957). *Parkinson's Law: Or the Pursuit of Progress (Penguin Modern Classics)*. New York: Buckaneer books.

Parsons, T. (1951). *The Social System.* (2nd ed.). London: Routledge & Kegan Paul Ltd.

Paul, S., Samarah, I. M., Seetharaman, P., & Mykytyn, P. P. J. (2005). An Empirical Investigation of Collaborative Conflict Management Style in Group Support System-Based Global Virtual Teams. *Journal of Management Information Systems*, *21*(3), 185–222.

Paul, S., Seetharaman, P., Samarah, I., & Mykytyn, P. P. (2004). Impact of heterogeneity and collaborative conflict management style on the performance of synchronous global virtual teams. *Information and Management*, *41*(3), 303–321.

Pazos, P. (2012). Conflict management and effectiveness in virtual teams. *Team Performance Management*, *18*(7), 401–417.

Pearce, C. L. (2006). The future of leadership: combining vertical and shared leadership to transform knowledge work. *IEEE Engineering Management Review*, *34*(1), 47–59.

Pearce, J. P. (1993). *Volunteers: the organizational behavior of unpaid workers*. London: Routledge.

Peterson, R. A., & Kim, Y. (2013). On the relationship between coefficient alpha and composite reliability. *The Journal of Applied Psychology*, *98*(1), 194–8.

Pinkley, R. L. (1990). Dimensions of conflict frame: Disputant interpretations of conflict. *Journal of Applied Psychology*, *75*(2), 117–126.

Polzer, J. T., Crisp, C. B., Jarvenpaa, S. L., & Kim, J. W. (2006). Extending the faultline model to geographically dispersed teams: How colocated subgroups can impair group functioning. *Academy of Management Journal*, *49*(4), 679–692.

Pondy, L. R. (1967). Organizational Conflict: Concepts and Models. *Administrative Science Quarterly*, *12*(2), 296–320.

Poole, M. S., & Roth, J. (1989, mar). Decision Development in Small Groups IV A Typology of Group Decision Paths. *Human Communication Research*, *15*(3), 323–356.

Poole, M. S., Van De Ven, A. H., Dooley, K., & Holmes, M. E. (2000). *Organizational Change and Innovation Processes: Theory and Methods for Research*. New York: Oxford University Press.

Postmes, T., Postmes, T., Spears, R., Spears, R., Lea, M., & Lea, M. (2000). The formation of group norms in computer-mediated communication. *Human Communication Research*, *26*(3), 341–371.

Postmes, T., Spears, R., & Lea, M. (1998). Breaching or Building Social Boundaries? SIDE-Effects of Computer-Mediated Communication. *Communication Research*, *25*(6), 689–715.

Puranam, P., Alexy, O., & Reitzig, M. (2014). What's "New" ABout New Forms of Organizing. *Academy of Management Review*, *39*(2), 162–180.

Purvanova, R. K., & Bono, J. E. (2009). Transformational leadership in context: Face-to-face and virtual teams. *Leadership Quarterly*, *20*(3), 343–357.

Putnam, L. L., & Poole, M. S. (1987). Conflict and Negotiation. In *Handbook of organizational communication: An interdisciplinary perspective* (pp. 549–599).

Putnam, L. L., & Stohl, C. (1990). Bona fide groups: A reconceptualization of groups in context. *Communication Studies*, *41*(3), 248–265.

Qiu, J., Wang, C., & Cui, M. (2013). The influence of cognitive conflict on the result of collaborative editing in Wikipedia. *Behaviour & Information*

Technology, *33*(12), 1361–1370.

Qureshi, I., & Fang, Y. (2011). Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. *Organizational Research Methods*, *14*(1), 208–238.

Raes, E., Kyndt, E., Decuyper, S., Van den Bossche, P., & Dochy, F. (2015). An Exploratory Study of Group Development and Team Learning. *Human Resource Development Quarterly*, *26*(1), 5–30.

Rahim, M. A. (2002). Toward a Theory of Managing Organizational Conflict. *International Journal of Conflict Management*, *13*(3), 206–235.

Rails Guides. (2014).Retrieved from [2014-07-25]`http://guides.rubyonrails.org/getting{_}started.html`

Rapoport, A. (1960). *Fights, games, and debates*. University of Michigan Press.

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49.

Reicher, S. D., Spears, R., & Postmes, T. (1995). A Social Identity Model of Deindividuation Phenomena. *European Review of Social Psychology*, *6*(1), 161–198.

Robles, G., Scheider, H., Tretkowski, I., & Weber, N. (2001). *Who is doing it? A research on libre software developers* (Tech. Rep.).

Roccas, S., & Brewer, M. B. (2002). Social Identity Complexity. *Personality and Social Psychology Review*, *6*(2), 88–106.

Rutkowski, A.-F., Saunders, C., Vogel, D., & van Genuchten, M. (2007). "Is It Already 4 a.m. in Your Time Zone?": Focus Immersion and Temporal Dissociation in Virtual Teams. *Small Group Research*, *38*(1), 98–129.

S, H. (2016). *Comment on "Code of Conduct"*.Retrieved from [15-2-2016]`https://bugs.ruby-lang.org/issues/12004{#}note-122`

Saavedra, R., Earley, P. C., & Van Dyne, L. (1993). Complex interdependence in task-performing groups. *Journal of Applied Psychology*, *78*(1), 61–72.

Saunders, C. S., & Ahuja, M. K. (2006). Are All Distributed Teams the Same? Differentiating Between Temporary and Ongoing Distributed Teams. *Small Group Research*, *37*(6), 662–700.

Schlueter, I. (2014). *Comment on "The Next Phase of Node.JS."*.Retrieved from [2014-05-01]`https://news.ycombinator.com/item?id=7064470`

Sen, R., Singh, S. S., & Borle, S. (2012). Open source software success: Measures and analysis. *Decision Support Systems*, *52*(2), 364–372.

Sharp, S. (2013). *Re: [00/19] 3.10.1-stable review*.Retrieved from `http://marc.info/?l=linux-kernel{&}m=137390362508794{&}w=2`

Sharp, S. (2015). *Closing a door.*Retrieved from [2015-10-01]http://sarah.thesharps.us/2015/10/05/closing-a-door/

Sherif, M. (1936). *The psychology of social norms*. New York: Harper & Brothers.

Shin, Y. (2005). Conflict resolution in virtual teams. *Organizational Dynamics*, *34*(4), 331–345.

Short, J., Williams, E., & Christie, B. (1976). *The Social Psychology of Telecommunications.* London: John Wiley & Sons.

Simmel, G. (1955). *Conflict , translated by Kurt H. Wolff. The web of groupaffiliations, translated by Reinhard Bendix.* Glencoe, IL: Free Press.

Simons, T. L., & Peterson, R. S. (2000). Task conflict and relationship conflict in top management teams: the pivotal role of intragroup trust. *The Journal of applied psychology*, *85*(1), 102–111.

Sobel, M. E. (1982). Asymptotic intervals for indirect effects in structural equations models. In S. Leinhart (Ed.), *Sociological methodology* (pp. 290–213). San Francisco: Jossey-Bass.

Somech, A. (2006). The Effects of Leadership Style and Team Process on Performance and Innovation in Functionally Heterogeneous Teams. *Journal of Management*, *32*(1), 132–157.

Spears, R., & Lea, M. (1992). Social influence and the influence of the 'social' in computer-mediated communication. In M. Lea (Ed.), *Contexts of computer-mediated communication* (pp. 30–65). England: Harvester Wheatsheaf.

Spears, R., & Lea, M. (1994). Panacea or Panopticon? The Hidden Power in Computer-Mediated Communication. *Communication Research*, *21*(4), 427–459.

Sproull, L., & Kiesler, S. (1986). *Reducing Socia Context Cues: Electronic Mail in Organizational Communication* (Vol. 32) (No. 11).

Staff, C. (2006). *Torvalds lambasts Free Software FoundationTorvalds lambasts Free Software Foundation.*Retrieved from [2012-10-01]http://news.cnet.com/8301-10784{_}3-6099985-7.html

Stallman, R. M. (2002). *Free Software Free Society: Selected Essays of Richard M. Stallman* (J. Gay, Ed.). Boston, MA: GNU Press.Retrieved from http://www.gnu.org/doc/fsfs-ii-2.pdf

Staples, D. S., & Zhao, L. (2006). The effects of cultural diversity in virtual teams versus face-to-face teams. *Group Decision and Negotiation*, *15*(4), 389–406.

Stark, E., Bierly†, P., & R. Harper, S. (2014). The interactive influences of conflict, task interdependence and cooperation on perceptions of virtualness

in co-located teams. *Team Performance Management: An International Journal*, *20*(5/6), 221–241.

Stewart, G. L. (2006). A Meta-Analytic Review of Relationships Between Team Design Features and Team Performance. *Journal of Management*, *32*(1), 29–55.

Stewart, K. J., & Gosain, S. (2006). The Impact Of Ideology On Effectiveness In Open Source Software Development Teams. *MIS Quarterly*, *30*(2), 291–314.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research: grounded theory procedure and techniques*. London: Sage.

Subramaniam, C., Sen, R., & Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, *46*(2), 576–585.

Tajfel, H. (1978). Social categorization, social identity and social comparison. In *Differentiation between social groups: Studies in the social psychology of intergroup relations* (pp. 61 – 76).

Tannenbaum, S. I., Mathieu, J. E., Salas, E., & Cohen, D. (2012). Teams Are Changing: Are Research and Practice Evolving Fast Enough? *Industrial and Organizational Psychology*, *5*(1), 2–24.

Taylor, F. W. (1911). *The principles of scientific management*. New York: Harper & Brothers.

Tekleab, A. G., Quigley, N. R., & Tesluk, P. E. (2009). A Longitudinal Study of Team Conflict, Conflict Management, Cohesion, and Team Effectiveness. *Group & Organization Management*, *34*(2), 170–205.

Thatcher, S., Jehn, K. A., & Zanutto, E. (2003). Cracks in Diversity Research: The Effects of Diversity Faultlines on Conflict and Performance. *Group Decision and Negotiation*, *12*(3), 217–241.

Thomas, K. W. (1992). Conflict and conflict management: Reflections and update. *Journal of Organizational Behavior*, *13*(3), 265–274.

Thomas, K. W., & Pondy, L. R. (1977). Toward an "intent" model of conflict management among principal parties. *Human Relations*, *30*(12), 1089–1102.

Thomas, K. W., & Schmidt, W. H. (1976). A Survey of Managerial Interests with Respect to Conflict. *Academy of Management Journal*, *19*(2), 315–318.

Towne, W. B., Kittur, A., Kinnaird, P., & Herbsleb, J. (2013). Your process is showing: Controversy Management and Perceived Quality in Wikipedia. In *Proceedings of the 2013 conference on computer supported cooperative work - cscw '13* (p. 1059). San Antonio, TX: ACM Press.

Tuckman, B. W. (1965). Developmental Sequence in Small Groups. *Psychological bulletin*, *63*(6), 384–399.

Tullio, D., & Devan, J. (2008). Toward a Theory of Socialization in Open Source Software Communities : A Symbolic Interactionist Perspective. In *Proceedings of the fourteenth americas conference on information systems* (pp. 1–8). Toronto, ON, Canada.

Turner, J. C., Hogg, M. A., Oakes, P. J., Reicher, S. D., & Wetherell, M. S. (1987). *Rediscovering the Social Group: A Self-Categorization Theory.* New York: Basil Blackwell.

van Wendel de Joode, R. (2004). Managing Conflicts in Open Source Communities. *Electronic Markets*, *14*(2), 104–113.

Vasilescu, B., Filkov, V., & Serebrenik, A. (2015). Perceptions of Diversity on GitHub : A User Survey. In *Proceedings of the 8th international workshop on cooperative and human aspects of software engineering.* CHASE '15.

von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, *32*(7), 1217–1241.

W3Schools. (n.d.). *Browser Statistics.*Retrieved from [2015-09-01]`http://www.w3schools.com/browsers/browsers{_}stats.asp`

Wageman, R. (1995). Interdependence and group effectiveness. *Administrative Science Quarterly*, *40*(1), 145–180.

Wageman, R., Gardner, H., & Mortensen, M. (2012). The changing ecology of teams: New directions for teams research. *Journal of Organizational Behavior*, *33*(3), 301–315.

Wagner, C., & Prasarnphanich, P. (2007). Innovating collaborative content creation: The role of altruism and wiki technology. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 1–10.

Wakefield, R. L., Leidner, D. E., & Garrison, G. (2008). A model of conflict, leadership, and performance in virtual teams. *Information Systems Research*, *19*(4), 434–455.

Wall, J. A., & Callister, R. R. (1995). Conflict and Its Management. *Journal of Management*, *21*(3), 515–558.

Wall, V. D., & Nolan, L. L. (1986, nov). Perceptions of Inequity, Satisfaction, and Conflict in Task-Oriented Groups. *Human Relations*, *39*(11), 1033–1051.

Waller, M. J., Zellmer-Bruhn, M. E., & Giambatista, R. C. (2002, oct). Watching the clock: group pacing behavior under dynamic deadlines. *Academy of Management Journal*, *45*(5), 1046–1055.

Walther, J. B. (1996). Computer-Mediated Communication: Impersonal, Interpersonal, and Hyperpersonal Interaction. *Communication Research*, *23*(1), 3–43.

Walther, J. B. (1997). Group and Interpersonal Effects in International Computer-Mediated Collaboration. *Human Communication Research*, *23*(3), 342–369.

Walther, J. B., Anderson, J. F., & Park, D. W. (1994). Interpersonal Effects in Computer-Mediated Interaction: A Meta-Analysis of Social and Antisocial Communication. *Communication Research*, *21*(4), 460–487.

Walther, J. B., & Bazarova, N. N. (2007). Misattribution in Virtual Groups: The Effects of Member Distribution on Self-Serving Bias and Partner Blame. *Human Communication Research*, *33*(1), 1–26.

Walther, J. B., & Burgoon, J. K. (1992). Relational Communication in Computer-Mediated Interaction. *Human Communication Research*, *19*(1), 50–88.

Walton, R. E. (1969). *Interpersonal peacemaking: confrontations and third-party consultation*. Addison-Wesley.

Wang, J., Shih, P. C., & Carroll, J. M. (2015). Revisiting Linus's law: Benefits and challenges of open source software peer review. *International Journal of Human-Computer Studies*, *77*, 52–65.

Wasko, M. M., & Faraj, S. (2005). Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. *MIS Quarterly*, *29*(1), 35–57.

Weber, S. (2005). *The Success of Open Source*. Harvard University Press.

Windeler, J. B., Robert, L. P., & Riemenschneider, C. K. (2015). E-profiles, Conflict, and Shared Understanding in Distributed Teams. *Journal of the Association for Information Systems*, *16*(7), 608–645.

Worchel, S. (1994). You Can Go Home Again: Returning Group Research to the Group Context with an Eye on Developmental Issues. *Small Group Research*, *25*(2), 205–223.

Wu, J., Tang, Q., & Pacis. (2007). Analysis of Survival of Open Source Projects: a Social Network Perspective. In *Pacific asia conference on information systems 2007, sections 1-6* (pp. 55–68).

Xu, B., Jones, D. R., & Shao, B. (2009). Volunteers' involvement in online community based software development. *Information and Management*, *46*(3), 151–158.

Yang, J., & Mossholder, K. W. (2004). Decoupling task and relationship conflict: the role of intragroup emotional processing. *Journal of Organizational Behavior*, *25*(5), 589–605.

Yetton, P., & Sharma, R. (2007). The Contingent Effects of Training, Technical Complexity, and Task Interdependence on Successful Information Systems Implementation. *MIS Quarterly*, *31*(2), 219–238.

Yilmaz, G., & Pena, J. (2014). The Influence of Social Categories and Interpersonal Behaviors on Future Intentions and Attitudes to Form Subgroups in Virtual Teams. *Communication Research*, *41*(3), 333–352.

Zaheer, S., Albert, S., & Zaheer, A. (1999). Time Scales and Organizational Theory. *The Academy of Management Review*, *24*(4), 725.

Zhu, H., Kraut, R., & Kittur, A. (2012). Organizing without formal organization. In *Proceedings of the acm 2012 conference on computer supported cooperative work - cscw '12* (p. 935). Seattle, WA: ACM Press.

Appendices

.1 Inter-Item Correlations

Table 1: Inter-Item Correlations

Variables		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Affective Conflict	1	1.00										
Task Conflict	2	.47**	1.00									
Process Conflict	3	.46**	.32**	1.00								
Normtive Conflict	4	.50**	.50**	.45**	1.00							
Performance Perception	5	-.05	-.14*	-.21**	-.30**	1.00						
Intent To Remain	6	-.18**	-.20**	-.16*	-.32**	.23**	1.00					
Identification	7	-.09	-.08	-.15*	-.28**	.33**	.37**	1.00				
Interdependence	8	.07	.15*	.20**	.05	-.05	.00	.20**	1.00			
Team Distribution	9	.06	.05	-.08	-.06	.20**	.15*	-.02	-.13	1.00		
Leadership Style	10	-.31**	-.14	-.17*	-.31**	.35**	.34**	.29**	.06	.26**	1.00	
Decision-Making Distrib	11	-.17*	-.11	-.23**	-.25**	.19**	.26**	.43**	.07	-.08	.30**	1.00

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

.2 Literature examining conflict in virtual teams

Table 2: Studies examining conflict in virtual teams sorted by participant type and year

Study	Year	Variables	Method	Tenure	Participants
(Martins et al., 2004)	2004	Review of research on virtual teams	Review		
(Shin, 2005)	2005	Mapping virtuality dimensions to sources of conflict	Theory	-	-
(Curşeu, 2006)	2006	Virtual teams as complex socio technical systems	Theory	-	-
(Saunders & Ahuja, 2006)	2006	Differentiating between short term and long term oriented distributed teams	Theory	-	-
(Martins & Schilpzand, 2011)	2011	Overview of research on GVTs	Review	-	-
(Gilson et al., 2014)	2014	Overview of research on virtual work	Review	-	-
(Chidambaram et al., 1991)	1991	Conflict management, virtualness, Cohesion	Longitudinal Experiment	1 month, 4 related tasks/sessions	Students
(Cramton, 2001)	2001	Communication volume, information distribution, team performance	Content Analysis of conflict episodes	Not specified, 1 Task/Project	Students
(Montoya-Weiss et al., 2001)	2001	CMC effects on conflict management	Lab Experiment	2 Weeks, 1 Task/Project	Students
(Paul, Seetharaman, Samarah, & Mykytyn, 2004)	2004	Heterogeneity, Collaborative Conflict Management, Satisfaction, Performance,, Participation, Decision Quality	Lab Experiment	3 days,1 Task/Project	Students

Study	Year	Variables	Method	Tenure	Participants
(Paul et al., 2005)	2005	Collaborative Conflict Management, Diversity, Cultural Traits, Decision Agreement, Satisfaction	Lab Experiment	<1 day, 1 project with 3 related activities	Students
(Staples & Zhao, 2006)	2006	Virtuality, Heterogeneity, Conflict as single measure, Cohesion	Lab Experiment	1 day, 1 Task/Project	Students
(Kankanhalli et al., 2007)	2007	Task, Affective Conflict, Diversity, Virtuality, Task complexity/Interdependence, Conflict Resolution Style	Qualitative	5 months1 Task/Project	Students
(Rutkowski et al., 2007)	2007	Immersion, Temporal Disassociation, Performance, Interpersonal Conflict	Lab Experiment	8 weeks, 1 Project with related sequential activities	Students
(Furumo, 2008)	2008	Task, Affective Conflict, Trust, Cohesion, Active members, deadbeats, deserters	Lab Experiment	1 Week, 1 Task/Project	Students
(Lira et al., 2008)	2008	Task, Relationship Conflict, Team Potency, Virtuality	Longitudinal	1 month, 1 Project	Students
(O'Leary & Mortensen, 2010)	2010	Faultlines, Subgroup Imbalance, Coordination problems, Conflict as a single measure	Lab Experiment	3 Weeks, 1 Task/Project	Students
(Martinez-Moreno et al., 2012)	2012	Virtuality, Task, Relationship Conflict	Longitudinal Experiment	1 month, 4 sessions/intervals	Students
(Pazos, 2012)	2012	Goal Commitment, Conflict Management, Team Performance	Quasi-Experiment	13 weeks, 1 project	Students

Study	Year	Variables	Method	Tenure	Participants
(Ayoko et al., 2012)	2012	Team development cycle, conflict as a single measure	Qualitative	Total tenure not specified, 1 Project	Students
(Stark et al., 2014)	2014	Impact of relationship, process conflict and cooperation on subsequent team Virtuality	Survey/Cross-sectional	Not specified	Students
(Yilmaz & Pena, 2014)	2014	Subgroup formation, Biased information sharing, Conflict as a single measure, Social categorization, Interpersonal behavior	Lab Experiment	<1 day, 1 Task/Project	Students
(Dodoiu, 2015)	2015	Theory of planned behavior, intention to engage in cooperative conflict management	Survey/Cross-sectional	3 months prior to survey, 1 Project	Students
(Martinez-Moreno et al., 2015)	2015	Self-guided training, Virtuality, functional conflict management	Lab Experiment	< 1 day, 1 Task/Project	Students
(Windeler et al., 2015)	2015	E-profiles, Task, Relationship Conflict, Shared Understanding, Team Effectiveness	Lab Experiment	< 1 day, 1 Task/project	Students
(Mortensen & Hinds, 2001)	2001	Shared Identity, Task, Relationship Conflict, Virtuality	Survey	15 months (average)	Real Product development teams
(Hinds & Mortensen, 2005)	2005	Spontaneous Communication, Shared Identity, Task, Relationship Conflict, Virtuality	Survey	9 years (average)	Real Product development teams
(Bosch-sijtsema, 2007)	2007	Expectation conflicts and mismatch	Qualitative	3 years (project 1) and 1 year (project 2)	Real Organizational Teams

Study	Year	Variables	Method	Tenure	Participants
(Boone & Hendriks, 2008)	2008	Conflict due to Locus of Control differences	Interview and survey	Not specified	Real World Top Management Teams
(Jong et al., 2008)	2008	Virtuality, Task, Affective, Process Conflict, Performance	Survey/Cross-Sectional	13.3 months (average)	Real World Teams
(Wakefield et al., 2008)	2008	Leadership, Task, Relationship, Process Conflict, Performance	Survey/Cross-Sectional	Av 2.97 years	Real World Top Management Teams
(Campbell et al., 2009)	2009	Conflict as force for establishing group social order	Critical Ethnography	Ongoing	Real World Online Financial Community
(Han & Harms, 2010)	2010	Identification, Task, Relationship Conflict, Trust	Field surveys	9.64 years (Study 1)/6.21 years (Study 2)	Real World Fortune 500 and Healthcare organization teams
(Arazy et al., 2011)	2011	Cognitive diversity, task conflict, article quality	Content Analysis	Ongoing	Wikipedia
(Ortiz De Guinea et al., 2012)	2012	Meta-analysis of work on Virtuality	Review	Short term (<1 day) versus long term (> 1 day)	Real World and Student Teams
(Arazy et al., 2013)	2013	Task, Task-Affective, Task-Process Conflict, Article Quality	Content Analysis	Ongoing	Wikipedia

.3 Survey Item Measures

The survey item measures used for Study 2 can be found below. Participants were instructed to answer questions from the perspective of the project they are currently most actively participating in. "This project" refers to participants most active project.

Distribution of decision-making (DECIS) (Campion et al., 1996)

Contributors are responsible for determining the methods, procedures, and schedules with which the work gets done

Contributors, rather than the project leader/founder, decide who does what task within the project

Most work related decisions are made by other contributors rather than the project leader/founder

As a contributor to this project, I have a real say in how the group carries out its work

Most contributors to this project get a chance to participate in decision-making

This project is designed to let everyone participate in decision-making

Leadership style (LEAD) (Carless et al., 2000)

Communicates a clear and positive vision of the future of this project

Treats contributors as individuals, supports and encourages their development

Gives encouragement and recognition to contributors

Fosters trust, involvement and cooperation among contributors

Encourages thinking about problems in new ways and questions assumptions

Is clear about his/her values and practices what he/she preaches

Instills pride and respect in others and inspires me by being highly competent

Task Interdependence (INTER) (Yetton & Sharma, 2007)

My work can be performed fairly independently of others (reversed)

My work can be executed/planned with little need to coordinate with others (reversed)

It is rarely required to obtain information from others to complete my work (reversed)

My work is relatively unaffected by the performance of other members (reversed)

My work requires frequent coordination with the effort of others

My performance on tasks is dependent on receiving accurate information from others

Team Distribution (DISTRIB) (Chudoba et al., 2005)

I collaborate with people in different time zones

I work with people via Internet-based communication applications (e.g. mailing lists, IRC, forums)

I collaborate with people I have never met face-to-face
I collaborate with people who speak different native languages or dialects than mine

Intention to Remain (ITR) (Jehn, 1995)

If I have my own way, I will continue working on this project
I do not expect to stay in this project very much longer (reversed)
I have thought seriously about leaving this project (reversed)

Performance Perception (PERF) (McDonough et al., 2001)

Attaining the goals set for this project
Bringing features to users rapidly
Developing features that are successful with users
Developing high quality features
Meeting user needs
Overall performance

Identification (IDENT) (Greene, 2014)

When someone criticizes this project, it feels like a personal insult
I'm very interested in what others think about this project
When I talk about this project, I usually say we rather than they
I have a number of qualities typical of members of this project
This projectâs successes are my successes
When someone praises this project, I feel like it is a personal compliment

Affective Conflict (AFFECT) (Jehn, 1995)

There is friction among contributors
Personality conflicts are evident
There is interpersonal tension among contributors
There are emotional displays (i.e. swearing, flaming, capitalization) among contributors

Task Conflict (TASK) (Jehn, 1995)

Contributors have different opinions regarding the work being done
There are differences of opinion in this project
There is conflict among contributors about the work being done
There are there conflicts about ideas in this project
There are differences of opinion about what features should be added
Contributors disagree on proposed changes to this project's code base
Contributors disagree about coding style

Process Conflict (PROC) (Jehn et al., 1999)

There are disagreements about who should do what in this project
There is conflict among contributors about task responsibilities
Contributors disagree about resource allocation
There are disagreements about authority in this project

Normative Conflict (NORM)

Contributors to this project disagree about the project mission
There are disagreements concerning the values this project upholds
Contributors disagree about what is expected behavior in this project
Contributors disagree about the tone and style of communication used within the group
There are ideological disagreements
Contributors disagree about conventions used in this project
Contributors debate the agreed upon customs and norms of this project

Roughly, how many people contribute to this project on a regular basis?

I'm the only contributor
2-10
11-20
21-30
31-40
41-50
More than 50
I'm not sure

Which of the following best describes how you think of yourself?

Male
Female
In another way: [please specify]

How long have you been contributing to this project?

---

1 year or less
About 2 years
About 3 years
4 or more years

On average, how much time per week do you spend on this project?
please specify
Hours per week

Please select the area where your main contributions to this project are:

(Please choose only one of the following)

Code, Programming
Other Contributions (e.g. documentation, translation, tests, artwork)
Both

In this project, I am

---

the leader/founder
a core developer/maintainer
a member with commit access
I do not have an official title
Other (please specify)