Social Integration in Agile User Experience: 
Building Social Capital in Agile User Experience Software Teams

Jeremy Totton Barksdale

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

D. Scott McCrickard, Chair
James D. Arthur
Jamika D. Burge
Edward A. Fox
Robin H. LeMaire

July 12, 2013
Blacksburg, VA

Keywords: Agile Usability, Agile User Experience, User Experience Knowledge Management, Social Capital, Network Governance, Qualitative Comparative Analysis (QCA), Social Network Analysis (SNA)

Copyright © 2013, Jeremy T. Barksdale
Social Integration in Agile User Experience: Building Social Capital in Agile User Experience Software Teams

Jeremy Totton Barksdale

Abstract

As the practice of software engineering matures, project teams are leveraging the expertise of those with a background in other disciplines such as user experience. This multidisciplinary collaboration has implications on how user experience is incorporated into the software they produce. It also has consequences for the interaction within the team. This research aims to address the implications and consequences by explaining and evaluating the impact of socio-cognitive factors and governance forms on agile user experience software teams. The objective is to support multidisciplinary agile user experience software teams in managing their interaction as a means to improving how user experience knowledge is managed. Results from a qualitative comparative analysis (QCA) were that: a combination of trust and shared meaning are associated with the impediment of knowledge construction and dissemination; a combination of lead governance, trust, and shared meaning are associated with knowledge dissemination; and a combination of lead governance and shared meaning are associated with the impediment of knowledge use. Review from an expert review of the Team Interaction Framework were that there are benefits to using the framework and ways to ease it use, but also limitations and anticipated challenges to its application. The findings from this research suggest that each theoretical component of the framework is relevant, but it is unclear whether the structural dimension is useful when studying agile user experience teams given environment similarity across teams. The contributions of this research are the Team Interaction Framework as a guide to evaluating the social interaction in agile user experience teams, a method for assessing the social interaction in agile user experience teams via a Team Interaction Assessment, and lightweight practices for improving the social interaction in these teams.
To my family, who has supported my lifelong journey and goals. I am where I am, and who I am, because of your love and investment over the years. I, especially, dedicate this work to those who have invested the most in my life: my mom, who passed too soon; Papa, who passed just as I started this journey; and Nana, who has been cheering me on throughout this journey.
Many have contributed to my successful completion of this work and chapter in my life. As I was writing my acknowledgments, I realized just how many people played a role in my getting to this point. I am grateful for every bit of the support and take none of it for granted. Although it is difficult to mention everyone by name, I deeply appreciate everyone's contribution.

I thank my advisor—Scott McCrickard—for his time and effort throughout the process. Being a good advisor requires a lot of patience and selflessness. A good advisor should be your strongest advocate, yet challenge you to think broadly and deeply. I was fortunate to have an advisor with these qualities, who provided the flexibility to let my research grow naturally. Might I add that he even supported my interest in simultaneously pursuing the Master's in Public Administration (MPA): what probably seemed loosely or unrelated at first. Scott, I appreciate your encouragement, support, and commitment to my successful completion of the Ph.D. program and other goals.

I also thank my committee members who have provided guidance and support throughout the process. I can only imagine (and will probably find out one day) what it's like to sit through the various defenses, and I appreciate your collective wisdom and feedback that helped to improve the quality of my work. I wanted a committee that represented my dissertation work, but I wanted it to be comprised of supportive and encouraging members. Each of you significantly contributed to the respective components of my dissertation, and I'm honored that you all served on my committee. Thank you, Dr. Arthur and Dr. Fox, for bringing your years of experience in science. Jamika and Robin, thank you for the additional words of comfort and lending your ear during the difficult times. They were more valuable than you can imagine.

Having a supportive University, Graduate School, and Department are imperative. I have enjoyed my time at Virginia Tech and that is in large part due to the welcoming climate of the University, a great Graduate School (that will get your paperwork through in record time), and an extremely supportive Department of Computer Science. Collectively, they have shielded me from major administrative and financial burdens: enabling me to focus my attention on growth as a researcher and a professional. I have to give special mention to Dr. Pérez-Quiñones for his advocacy during my time at Virginia Tech, and his (and Marilyn Kershaw's) role in my obtaining the Southern Regional Educational Board (SREB) Fellowship.
I had a great cadre of friends from my journey prior to Virginia Tech, at Virginia Tech in the Computer Science Department, in the Center for Public Administration and Policy (CPAP), and from my Brothers of Omega Psi Phi Fraternity, Inc. I thank my close friends—that I knew prior to entering Virginia Tech—who have contributed to my doctoral experience by providing a listening ear, advice, intellectual stimulation, or ongoing encouragement. I thank Laurian Vega and Adam Pickeral for their help with the studies. I also thank Shahtab for the thoughtful discussions in the lab and help while I was job hunting. There are too many to thank in CPAP for the fun times at the cabin, PK’s, TCH, and all the other good times that made my time in CPAP well worth it. I also thank the CPAP staff and faculty for their friendliness and thoughtful discussions in or out of class. I also thank my fraternity brothers (of the Eta Lambda chapter and brothers from a number of other chapters that live in the New River Valley) for making me feel welcome while at Virginia Tech.

In addition to the academic environment, I had a great peer community. Through becoming an SREB Doctoral Scholar, participation in the Empowering Leadership Alliance (ELA) and CHI Mentoring (CHIMe) Workshop, and through attending the ACM Richard Tapia Celebration of Diversity in Computing Conference, I met other Ph.D. students that would not just provide peer support, but with whom I would establish lasting friendships. These conferences have also helped with establishing mentoring relationships with people like Ron Metoyer, who would check in periodically to see how I was doing. These conferences, and the relationships that stemmed from them, were invaluable to my successful completion of the Ph.D. program. I look forward to repaying the favor by volunteering at conferences so other students can have the same benefit.

Internships have played a key role in my professional development as a researcher and practitioner. I owe a debt of gratitude to Mary Czerwinski, Kori Inkpen, and many others at Microsoft Research for the rewarding internship experience. I appreciate the opportunity to work with you, the rest of our research team—Asta, Paul, Aaron, Gina—as well as the other interns, study participants, and staff that made the research and publication possible. I had the pleasure of gaining practical UX experience at Meridium through working with a great team of people that allowed me to gain meaningful experience. Many thanks to Jason Lee for his role in helping me to secure the internship as well as his mentoring while at Meridium (and in the Ph.D. program, generally). Finally, I thank Terry Davis for helping me secure the internship at IBM, where I also gained additional practical experience that has contributed to my professional development. I am thankful for all of these internship experiences and will cherish them for years to come.
I am also appreciative to have had such great participants that were more than willing to allow me into their organization to collect data. I wish I could thank them by name, but we have to honor confidentiality and anonymity promises.

Finally, I thank my family for their encouragement and support through the years. Many of my family members have served as role models— influencing key decisions I would make in life when I didn't even realize it. Who knew those long and engaging conversations with my uncles, aunts, cousins, and grandparents would lead to a desire to pursue a Ph.D. I am grateful for having such a great family.
# Table of Contents

Chapter 1: Introduction .................................................................................................................. 1

  1.1. Motivation .......................................................................................................................... 2
  1.2. Research Problem ............................................................................................................. 3
  1.3. Thesis Statement ............................................................................................................... 4
  1.4. Research Purpose and Goals ............................................................................................. 4
  1.5. Outline of Chapters .......................................................................................................... 5

Chapter 2: Background .................................................................................................................. 8

  2.1. Agile Software Development .............................................................................................. 9
  2.2. Software User Experience ............................................................................................... 10
  2.3. Agile User Experience ...................................................................................................... 11
  2.4. Chapter Summary ............................................................................................................ 12

Chapter 3: Related Work .............................................................................................................. 13

  3.1. Literature Review Methodology ....................................................................................... 14
  3.2. Practices Integration ....................................................................................................... 16
  3.3. Process Integration .......................................................................................................... 18
  3.4. Technology Integration .................................................................................................... 18
  3.5. People Integration ........................................................................................................... 19
  3.6. Social Integration ............................................................................................................ 20
  3.7. Chapter Summary ............................................................................................................ 21

Chapter 4: Team Interaction Framework .................................................................................... 22

  4.1. Social versus Cognitive Theories ..................................................................................... 23
  4.2. Social Capital .................................................................................................................. 23
  4.3. Network Governance ....................................................................................................... 25
  4.4. Knowledge Management ................................................................................................. 28
References .......................................................................................................................... 83
Appendices Summary ........................................................................................................ 92
Appendix A: Analysis Details .............................................................................................. 93
    Data Table: Conditions .................................................................................................. 93
    Data Table: Outcomes ................................................................................................. 93
Appendix B: Team A Case Study ....................................................................................... 94
Appendix C: Team B Case Study ....................................................................................... 108
Appendix D: QCA Primer ................................................................................................ 124
    Introduction .................................................................................................................. 124
    Qualitative Comparative Analysis .............................................................................. 124
    Example Application ................................................................................................... 126
    Discussion ................................................................................................................... 129
    Conclusion .................................................................................................................. 131
Appendix E: Team Interaction Assessment ...................................................................... 132
Appendix F: IRB Documentation ...................................................................................... 140
List of Figures

Figure 1: Distribution of agile usability literature across themes................................................................. 15
Figure 2: Team Interaction Framework .......................................................................................................... 28
Figure 3: Chain of evidence............................................................................................................................ 41

List of Tables

Table 1: Agile usability integration themes, key authors, and related work summary ........................................ 15
Table 2: Definitions of Social Capital............................................................................................................... 24
Table 3: Key differences between statistical, QCA, and case analytical methods ............................................ 32
Table 4: Study phases ........................................................................................................................................ 35
Table 5: Summary of the conditions, definitions, and measures of each condition ........................................ 36
Table 6: Case summary results ...................................................................................................................... 47
Table 7: Knowledge Construction QCA Solution............................................................................................. 50
Table 8: Knowledge Embodiment QCA Solution ............................................................................................. 51
Table 9: Knowledge Dissemination QCA Solution .......................................................................................... 53
Table 10: Knowledge Use QCA Solution ......................................................................................................... 54
Table 11: QCA summary and impact table ...................................................................................................... 55
Table 12: Final truth table before running the standard analysis ..................................................................... 128
Chapter 1: Introduction

Chapter Outline:
- Motivation
- Problem Statement
- Thesis Statement
- Research Purpose and Goals
- Outline of Chapters

After reading this chapter, you should:
- Know the motivation for this research
- Understand the problem being addressed by this research
- Understand the context in which the problem is situated
- Know the purpose, goals, and focus of this research
Traditional software teams, comprised of software developers using waterfall-like methodologies, were limited in their ability to develop quality software on schedule and within budget. This was especially true on larger projects where more rigorous methods were required to satisfy governmental contractual obligations, or where it was necessary to accommodate standards such as the Capability Maturity Model Integration (CMMI) [27].

As the practice of software engineering matured, software teams began leveraging the knowledge and skills of experts in other areas, such as usability engineering and interaction design. This diversity of expertise allowed teams to delegate detailed attention needed for software quality attributes that might otherwise receive superficial consideration. An advantage of a diverse team is that they can produce better quality software in a shorter time period given the decreased time required by experienced members in each area and their parallel work activity. However, this diversity can come with challenges.

1.1. Motivation

Multidisciplinary software teams can face the challenge of effectively transferring knowledge within the team. Encoding and decoding during knowledge transfer is complicated because members are less likely to have a shared vocabulary and meaning. For example, software developers and user experience (UX) professionals have a different understanding of what constitutes a scenario. A scenario for a software developer will most likely include a formal narrative of the user’s activity relative to the system; the UX professional’s scenario will likely take on a less formal narrative form that resembles a story of only the user’s actions. Although the two disciplines use the term "scenario," there is just enough similarity and difference in the meaning to interfere with gaining the shared understanding necessary for effective collaboration. Hence, the existence of similarities and differences adds to the complexity of a team’s interaction.

Such collaboration barriers can compromise the negotiation among members when making system and user experience decisions. Agile teams constantly prioritize which aspects of the software will get development effort. Given that developers are responsible for implementation, if negotiation breaks down between the two groups (e.g., due to miscommunication), it could result in the developers bypassing the input of user experience professionals and potentially compromising the product’s user experience and degree of innovation.
Previous research has investigated methods for improving agile user experience interaction. Such methods include technical process integration (e.g., eXtreme Scenario-based Design), the sharing of practices (e.g., daily stand-up meetings), and technology integration (e.g., design and development tools). Although there are benefits to these approaches, focus is needed on socio-cognitive interactions. This research aims to provide insight about how social capital and network governance contribute to the management of user experience knowledge on agile user experience software teams. It also seeks to offer guidance on designing cohesive agile user experience teams. It is anticipated that the awareness from this research will inform teams on ways to modify their dynamics toward improving how their user experience knowledge is managed.

1.2. Research Problem

The loyalty of team members to their discipline can mask the benefits experts from other disciplines bring to the team. Typically, loyalty to one’s discipline can be advantageous. It can create a cohesive environment within teams. However, it can also result in the formation of cliques that neglect team members from other disciplines. For example, if software developers neglect user experience professionals because they don’t share the same background, then user experience ideas may not get incorporated into the product. Conversely, if user experience professionals neglect software developers for the same reason, then software developers may not receive important user insight. This self-defeating behavior can adversely influence how user experience knowledge is managed, and ultimately, the product’s quality. The more practical problem is that the company may experience a loss in sales or reputation. Existing literature does not directly explain the impact of social capital and network governance on team interaction and user experience knowledge management.

Simply because developers and user experience professionals have processes, practices, and technology to help them collaborate doesn’t necessarily mean they will consistently employ them. Indeed, there are socio-cognitive factors that influence their motivation for adopting the idea of integrating their tasks. For example, if developers and user experience professionals don’t have a relationship with one another, they may ultimately choose to work in isolation (i.e., without the necessary interaction such as meetings) out of a preference for simplicity. This isolation could result in subtle, yet significant, user experience recommendations being dismissed or overlooked.
Hence, this research addresses the problem that:

*Agile user experience teams can experience social interaction challenges that can interfere with how user experience knowledge is managed thus, potentially, increasing the difficulty of maintaining agility and adversely affecting the quality of the product. Guidance is limited on how teams can improve knowledge management via strategic improvements to the social interaction between developers and designers.*

1.3. Thesis Statement

Given the emphasis on the effects of social interaction on knowledge management this research posits that:

*Social interactions on multidisciplinary agile user experience software teams—characterized as social capital factors and network governance forms—can influence how user experience knowledge is managed.*

To establish support for this argument, a mixed methods study was conducted—measuring social capital, assessing network governance form(s), and analyzing these elements relative to how user experience knowledge is managed in agile user experience software teams.

1.4. Research Purpose and Goals

Hence, the purpose of this research is to (1) provide researchers and practitioners with an understanding of how social capital and network governance influence the interaction between software developers and user experience professionals, and (2) provide guidance to practitioners via practices for how to strategically change how user experience knowledge is managed in their teams.

To fulfill this purpose, the goals of this research are to:

1. Explore the nature of social capital and network governance in agile user experience teams;
2. Provide possible explanations of the relationship between social capital, network governance forms, and user experience knowledge management;
3. Provide insight on the role of cohesion among software developers and user experience professionals on agile user experience teams; and
4. Develop a Team Interaction Framework to assist practitioners with designing and managing the social interaction in agile user experience teams.

1.5. Outline of Chapters

The dissertation is organized as follows to simplify reading and understanding its content. Each chapter is prefaced with an outline of sections in the chapter along with goals of what the reader should expect to learn from reading the chapter. Finally, most chapters contain a summary that reiterates the main points of the chapter. The appendices provide more detailed information about the study such as the data tables, case study reports, IRB forms, and the Team Interaction Assessment used to collect data.

Chapter 2. Background

The Team Interaction Framework is grounded in an understanding about agile development, user experience, and the motivation for the integration of agile and user experience. This chapter provides a foundational background of these concepts along with the rationale for why a Team Interaction Framework is important for agile user experience teams.

Chapter 3. Related Work

Various approaches to integrating agile and user experience have been proposed and used by researchers and practitioners. Understanding the history of agile user experience integration is important in establishing the rationale and context for the Team Interaction Framework. This chapter shares work that has significantly contributed to the integration of agile developers and user experience professionals. Specifically, this chapter presents the practices, process, technology, people, and social integration approaches.

Chapter 4. Team Interaction Framework

The Team Interaction Framework is used to guide the study of social interactions within agile user experience teams. This chapter defines the theoretical components of the framework—social capital, network governance, and knowledge management. This chapter also presents the goals and rationale of the framework. Finally, the chapter details how each of the theoretical components fit into the Team Interaction Framework.

Chapter 5. Research Design and Methods

The research design and methods used to address the overarching research question and thesis statement are presented in this chapter. Hence, this chapter includes the operational research questions and...
propositions. It also includes an overview of the comparative case study research design, qualitative comparative analysis (QCA), social network analysis (SNA), and the variables used in the study. Concerns of validity and reliability for case study research are detailed along with the other research method considerations (viz., materials, data collection, and data analysis). This chapter is important because it provides the core logic for this research.

Chapter 6. Results and Discussion
The results section of this chapter contains a summary of the results from the case studies, QCA results, and findings from the evaluation of the propositions. The results summary includes individual values from the analysis of each theoretical component of the Team Interaction Framework in each case study. The QCA results section provides insight about the relationship between the theoretical components. Findings from the evaluation of the propositions are informed by both case study and QCA results. Given the close relationship between the results and the cases, discussion is included in this same chapter to ease interpretation and understanding.

Chapter 7. Team Interaction Framework (Revisited)
An important step in developing the Team Interaction Framework is considering the relevance of each theoretical component in light of the findings. This chapter discusses the suitability of each component in the framework and the implications of the findings for practitioners. Furthermore, this chapter provides actionable guidance to agile user experience team members in the form of social interaction practices they can employ based on the findings from this research.

Chapter 8. Framework Application Guide
Practitioners typically don’t have the time to design and conduct a study to substantiate a framework. They usually prefer to have a framework they can apply with an understanding of the assumptions associated with the framework. This chapter provides practitioners a quick guide for using the framework to study the social interaction in agile user experience teams. It also provides insight about important aspects of the framework needed for practitioners to interpret the findings based on the experience from conducting the case studies in this research.

Chapter 9. Conclusions and Agenda
This concluding chapter notes the contributions and limitations of this research, and an agenda for future work.
Appendices

The appendices include tables with the raw and calibrated data that were used to perform the qualitative comparative analysis (QCA), reports from each case study, and introduction on the use of QCA in HCI, the Team Interaction Assessment questionnaire, and the IRB forms required to conduct the study.
Chapter 2: Background

Chapter Outline:

Agile Software Development
Software User Experience
Agile User Experience
Chapter Summary

After reading this chapter, you should:

● Understand what agile software development is and how it surfaced.
● Understand what user experience is, the roles in user experience, and its purpose.
● Understand the purpose of integrating agile software development and user experience.
An understanding of what agile user experience is and how it surfaced, the integration of agile development and user experience, and insight about the nature of these multidisciplinary software teams are important to appreciate the challenges these teams can experience. This section provides this foundation.

2.1. Agile Software Development

A meeting among 17 software experts in 2001 resulted in a manifesto that outlined a set of values and principles for developing software with agility as an alternative to the traditional development approaches [43]. Software developers and methodologists use the manifesto as a vision for how an agile team should function. Since its adoption by the development community, teams have taken various courses of action toward becoming agile. For example, some teams adopt the “spirit” of agility by employing a subset of the values and principles, some may rigorously apply agile principles and practices from inception, while others may transition into agile more progressively to mitigate organizational and team culture shock. These varying levels of adoption are acceptable given that different organizations and teams have different objectives and levels of need for agility.

A weakness of the traditional waterfall approaches was their inability to accommodate regular customer feedback, which affected the likelihood of delivering a product that served the client’s needs. Requirements were elicited at the beginning of the project during the requirements phase, and incorporating changes into the product based on customer feedback became increasingly infeasible as the development lifecycle progressed. Potential problems were that the client’s needs might have changed since the requirements gathering phase, the client was unable to sufficiently communicate their needs because they didn’t have a tangible reference point, or the software team simply misunderstood the client’s needs. As customers begin to see and experience the character of the software, they gain a clearer understanding of their needs and wants. This inability to flexibly adjust during development impacted how well the final product met the customers’ needs.

The traditional approaches also resulted in software projects that ran over budget or were not completed on schedule. For reasons similar to those previously mentioned (e.g., an inability to identify clear needs upfront), the software team could not accurately estimate the required cost and time it would take to develop a product. Essentially, the effort was too complex to fully estimate at the beginning of the project.
For example, teams had difficulty estimating the appropriate team size, the complexity of the algorithms, or the time to resolve development environment configuration management issues that arose. Each of these, and many other points of failure in a development project, erode the team’s ability to control software cost, quality, schedule, and scope [9]. Agile was one answer to addressing these quality, cost, and schedule challenges.

Agile approaches[52], such as eXtreme programming (XP) [9], Scrum [110], and others [4] established a vision for minimizing the amount of big upfront planning that was required for product development. For example, XP provides developers a set of rules, values, and practices [9]. In Scrum, emphasis is on the more general project management activities versus lower level development activities. Common to both, and other agile processes, is the incremental and iterative progression of product development. The scope of work is allocated across time such that the product is incrementally delivered over a number of iterations and release cycles. In doing so, the team is better able to: estimate development time, estimate associated costs, and accommodate change requests. Also, the customer is in a better position to provide feedback since they are able to envision the system incrementally throughout development. More opportunities are available for users to provide feedback since the development team deploys a working product at regular increments. On the surface, the advantages of agile appear to satisfy the major drawbacks of the traditional approaches. However, although the software experts that formulated agile had the best intentions, a key component in the development process was superficially considered—the user.

2.2. Software User Experience

The user has commonly been considered in some capacity in software engineering via design decisions with the growth of the Unified Modeling Language (UML) [12]. UML is an integrated work of three separate modeling languages developed for object oriented analysis and design (OOAD) [63]. Modeling the use of the system helps developers better decompose the system, eases development effort, and provides alternative views [62] of the system as its complexity increases. In UML, the user is considered via use case diagrams with actors and system components, use cases that detail the interaction between the actors and the system, and usage scenarios that list the tasks users need to accomplish. However, analysis of system use is not equivalent to the analysis of the system’s user experience, and emphasis was primarily on the use of the system (and, indirectly, the user). Artifacts were commonly developed by software experts with a system focus, in consultation with the customer, and portrayed the user as a
mechanical entity—compromising the ability to gain a richer awareness of the user’s experience and preferences.

With the emergence of human computer interaction (HCI) research over the past 40 years, user experience has grown into a discipline with its own methods and practices as a means for keeping the user in the forefront during software design and development. An understanding about users in computing increased (e.g., awareness about their mental models, cognitive load, tasks, etc.), which provided deeper insight about how software can best support their needs. Research also advanced insight about what is visually appealing to users and how software can become more pleasant to their senses. This progress resulted in the establishment of methods such as scenario-based design (SBD) [21], user experience design (UXD), user interface (UI) design, interaction design (IxD), usability engineering (UE), and user centered design (UCD). Depending on the team, the associated roles might include a visual designer that is concerned with establishing a visually appealing user interface, an Interaction (Ix) designer who designs the taskflow, or a UX researcher who studies whether users are able to effectively carry out their intended tasks efficiently. Collectively, people in these roles are categorically referred to as user experience professionals throughout this research given their emphasis on the user. Common across these approaches and roles is the research and design of what the user experiences—i.e., a greater focus on the user than the system and the advocating for users’ needs and desires during software design and development.

2.3. Agile User Experience

User experience progressively made its way into agile software teams. Software practitioners needed a way to satisfy customers through providing timely releases such that changes were feasible without compromising software quality, going over budget, and taking longer than scheduled. User experience professionals wanted (and needed) to be included in the process to ensure the software product is intuitive, appealing, and simple to use. The first need was addressed with the establishment of agile. The second need was satisfied as software developers realized the importance of placing greater emphasis on the user. The result was agile user experience.

Constantine was among the first, in 2002, to identify that the agile manifesto did not directly consider usability. He argued for the incorporation of usability through his usage-centered design approach—a card-based modeling and decision making process [28,30]. From this point onward, research continued
to seek ways to integrate user experience into agile environments. In some cases, the goal was to increase the software developer's attention on the user by providing the software team with relevant principles and guidelines. In other cases, the argument was made for a separate user experience role (or team) on the software project.

Although user experience is important in traditional software teams, the focus on user experience in agile environments has special importance. Agile environments are fast-paced relative to traditional software environments, and poor team interaction can constrain an agile team from achieving and maintaining velocity. In such situations, team members may begin to give less consideration to important user experience ideas as a means to recovering velocity. Hence, improving the interaction between developers and user experience professionals in agile environments can help teams alleviate challenges that might otherwise impede maintaining agility while giving due consideration to users’ experiences.

2.4. Chapter Summary

This chapter has focused on providing the context around how agile user experience came to exist. Agile was created to address the limitations of the traditional—waterfall—methods. User experience was later emphasized to better advocate for the user in software development. It was almost natural that these two complementary ideals would join to facilitate the development of quality, on schedule, within budget, and usable software. The next chapter provides a more detailed account of the various agile user experience integration approaches that have been studied.
Chapter 3: Related Work

Chapter Outline:

Literature Review Methodology
Practices Integration
Process Integration
Technology Integration
People Integration
Social Integration
Chapter Summary

After reading this chapter, you should:

● Understand the various approaches taken to integrate agile and user experience domains
● Understand key findings from previous studies
● Understand how this research fits in with the previous integration approaches
The integration of agile and user experience is a relatively new research area. Since 2002, various approaches to integration have been discussed among researchers and practitioners alike. Based on findings from the literature review, research on agile user experience can be organized within five themes—the adoption of practices in one or both areas, the combination of agile and user experience processes, communication between technology used by each domain, incorporation of team members, and integration via social methods. Following is a brief overview of the methodology employed during the review of the agile user experience literature and insight about each theme. The phrase “agile usability” is used throughout this section given the recent shift from “agile usability” to “agile user experience” in the field.

3.1. Literature Review Methodology

Over the past decade, agile usability has grown into what can be narrowed to a few key directions. Table 1 shows the integration themes of agile usability relative to each year since its emergence around 2002 along with key authors and their publications. The “no data” category means that an integration category was indeterminable from the publication. The “none” category means that no integration was presented in the publication, which could result from the article discussing agile usability as a topic but not proposing a method of integration. The primary goal of this literature review was to identify the inception of agile usability and the various turns it has taken since that point. Hence, the review is not meant to be exhaustive, but a representation of the trends over time as a way to acknowledge past research on the topic and reveal an area of need for future research in agile usability.

In that light, a search was conducted on the search space of “agile usability software” in the computer science and business categories for each year between 2000 and 2010, inclusive. Google Scholar was used as the search engine to identify related work given its broad reach and coverage of the most relevant publication databases for this topic. The search returned over 5200 results and resulted in a review of over 65 key publications. Each publication was abstracted by extracting its title, authors, publication venue and year, relevance to the topic, evidence strength, integration category, integration approach, key argument(s)/summary, integration rationale, findings, future work, and any miscellaneous comments. Descriptive statistics were computed and a thematic analysis was conducted to better understand the gaps in the agile usability research area.
Figure 1 shows the distribution of integration types in the literature: practices and process integration account for over half of the literature found on agile usability integration. The least amount of literature discusses technology integration, and the people and social integration literature accounts for just less than a quarter of the agile usability integration literature.

**Figure 1: Distribution of agile usability literature across themes**

<table>
<thead>
<tr>
<th>Integration Type (%)</th>
<th>Key Authors with Related Work</th>
</tr>
</thead>
</table>
3.2. Practices Integration

Practice integration was found to be one of the most common approaches to agile usability integration. It is defined here as integration that occurs through the adoption of principles or practices from another field. An example of practices integration is when an existing agile process is supplemented with usability practices, but does not entail the complete merging of independent processes.

Agile Incorporating Usability Practices

One approach to integrating practices is to incorporate usability practices into agile methods and teams. In this case, practitioners augment their agile methods to include some of the important usability practices. Meszaros and Aston [82] argue that “Some Design Up Front provides better guidance to the development team and provides earlier opportunities for feedback.” They discuss their experience with incorporating usability testing into an XP project by building paper prototypes and conducting wizard-of-oz testing. The project manager and agile coach developed the paper prototypes, the business lead conducted the usability test sessions, and members of the development team acted as the computer and played the role of the help system or observed participants. They found that integrating the practices into

<table>
<thead>
<tr>
<th>Methods</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology (2.9%)</strong></td>
<td>Memmel, Bock, &amp; Reiterer 2008 [77]; Nunes 2008 [91]</td>
</tr>
<tr>
<td><strong>People (8.8%)</strong></td>
<td>McDonald &amp; Welland 2003 [75]; Ghosh 2004 [46]; McInerney &amp; Maurer 2005 [76]; Lievesley &amp; Yee 2006 [69]; Ferreira, Noble, &amp; Biddle 2007 [40]; Singh 2008 [112]</td>
</tr>
</tbody>
</table>
the team was easy, that usability testing ensured that all work was accounted for and prevented last minute essential scope creep, and that integration resulted in a significant reduction of usability rework.

**Usability Incorporating Agile Practices**

Another approach to practices integration is the incorporation of agile practices into usability methods. In this case, this entailed tailoring usability methods to become more agile-like. Sy [117] argues that by adjusting how UCD is conducted, the team was able to harness its power to the agile characteristics of speed, responsiveness, and high implementation quality. They tailored their approach to conducting usability tests, interviews, and contextual inquiry to fit within the constraints of the agile framework. Hence, although the usability methods were ultimately integrated into the agile framework, the usability methods were adjusted to incorporate agile practices by decreasing the time required for, and granularity of, usability investigations. This was later implemented by synchronizing the agile and UCD activities for efficiency. They found that the new agile UCD method produced better-designed products than the waterfall versions of the same techniques by narrowing the gap between evaluation and incorporation of changes.

**Mutual Sharing of Practices**

In the most collaborative sense, agile and usability methods incorporate practices from each other to move the two areas closer to each another. From an agile practitioner’s perspective, Ambler [4] argues that UX is important to software development, and that both agile software developers and UX professionals need to adjust for successful integration to work. He recommended that UX professionals go beyond UX in their skillset, become embedded in agile software development (ASD) teams, give agile approaches a chance, and start looking beyond XP when tailoring practices. His suggestions to the agile community were to learn UX skills, accept that usability is a critical quality factor, and adopt UI and usage style guidelines.

Parsons and others [95] offer strategies for incorporating key practices from each area into agile and usability methods based on their practical experiences. They argue that agile methods have not typically incorporated HCI and usability techniques and tools into their software development processes, and that it is possible to integrate certain practices from usability engineering into an agile approach. Among other strategies for mutual practices integration, they recommend using iterative development throughout the lifecycle, merge user scenarios with stories, and allow testing of the UI in the user context regularly throughout all phases. They also found that the following ensures high HCI quality: co-locating a UI expert
with the development team, letting the product manager-owned backlog control the development process, regular assessment of the application by an external HCI consultant, and making supportive design and technology choices.

3.3. Process Integration

The integration of agile and usability processes is another common approach to integrating agile and usability efforts. In this case, independent agile and usability processes are combined and synchronized to provide a single newly designed agile usability process. The key difference between practices and process integration hinges on whether two completely separate processes were merged into a single process (viz., process integration) or techniques were selectively adopted from a process (viz., practices integration).

Constantine [28,29] advocates for the use of his Usage-Centered Design (UC-D) process in coordination with agile methods such as XP. He argues that his method is a natural adjunct to an effective agile process (rather than an end-to-end development process) because it is card-based (like story card use in XP), lightweight, and iterative and incremental in nature. Patton [97] later instantiates Constantine’s UC-D approach in an agile environment, arguing that interaction design is a valuable component of any software development process and that it happens whether the intention is there or not. Patton found that although constant collaboration was exhausting, the team’s tacit knowledge was “irreplaceable.” He also noted that the UC-D task cards were useful during testing and other points of reference.

Alternatively, Lee and others [65] integrated the scenario-based design (SBD) process and XP. They argued that there is a need to understand the similarities and differences between XP and SBD as a means to addressing the integration problem. A comparative analysis of the core principles of XP and SBD lead them to the development of extreme scenario-based design (XSBD). They found that maintaining and collectively agreeing on a prioritized list of design goals will help to resolve conflicts, claims are effective for capturing design rationale, enforced and opportunistic synchronization is important, and a central design record (CDR)—used to support synchronization of activities—can help with developing a cohesive interface.

3.4. Technology Integration

Technology integration means that the underlying coordination between the agile development and usability engineering activities occurs through the use of technology. An example of technology
integration is when the designer and developer can perform their task using two independent software components or applications that communicate using a common data exchange format to integrate their output. Pyla [102] developed a project development environment—Ripple—to foster communication between the software engineering and usability engineering roles. He argued for the need of a connection between SE and UE lifecycles to support communication among roles given they have different levels of iteration and evaluation, different terminology, and requirements representation.

Industry solutions to technology integration entail the use of a declarative markup language that facilitates data exchange between the designer’s software and the developer’s software. For example, when the designer creates an artifact (such as a UI), the code is automatically written in the background. The developer can then incorporate that component directly into the system code they've written. Common examples include applications provided by Microsoft and Oracle. Microsoft utilizes an eXtensible Markup Language (XML) based language—called eXtensible Application Markup Language (XAML)—that is implemented by their Windows Presentation Foundation (WPF). Microsoft Expression Blend is used to create UI elements and animations [71], Microsoft Visual Studio is used for system development, and both utilize XAML as a data interchange format for UI development.

On a relatively smaller scale, Oracle provides similar functionality through the use of a JavaScript Object Notation (JSON) based declarative markup language—called JavaFX Data (FXD)—as the data interchange format. The JavaFX Production Suite is used, in conjunction with Adobe Illustrator and Photoshop, by designers to construct UIs. The Netbeans integrated development environment (IDE) is used by developers to develop system code [26]. Design artifacts can be exported and incorporated into the system code via FXD.

3.5. People Integration

People-focused approaches achieve integration by changing the team’s personnel or composition to obtain the required talent and skills. This typically means, for example, adding a designer to the team, but not necessarily specifying an integrated process or set of practices that will be used. Singh argues that having two product owners in Scrum—called U-Scrum, where one focuses on agile while the other focuses on usability—can improve product usability. She found improvements in developer productivity with this structure [112]. She noted five factors that are critical to prevent potential obstacles with this approach—the two product owners should be peers, additional coordination may be required, an argument may be
needed to justify the additional role, the development team must view personas as an input to the process versus artificial creations, and the UX vision must provide a complete picture for the project.

McDonald and Welland [75] argue that creating agile multidisciplinary sub-teams of diverse specialists along with coordination teams to maintain communication among specialists will improve the inherent need for diversity in agile web engineering. They identify several stakeholder roles required for large web engineering projects. Namely, these are end-users, clients, domain experts, business experts, software engineers, creative designers, and team leaders. In their approach, sub-teams will include a number of different specialists as well as the infrastructure required for specialists to communicate across teams based on their specialty.

3.6. Social Integration

Social integration involves a remedy where integration occurs via the social construction of knowledge or changing how the team interacts socially. Ungar [120,121] provides an example of social integration through presentation of a design studio. He argues that holding a design studio (i.e., a workshop) with designers and developers is a viable approach to moving design ahead of development under the compressed time frame of Scrum. The design studio has four components: conduct user research, generate designs rapidly, evaluate created designs (i.e., the studio component), and use participants from various disciplines. The author found that the design studio facilitates role sharing and knowledge transfer, rapid exploration, early commitment, shared understanding, team cohesion, and the sharing of best practices.

Similarly, Brown and others [15] have investigated the role that stories, sketches, and lists (e.g., a product backlog) play in mediating the interaction between developers and designers. They argue that stories and sketches, as mediating artifacts, have critical roles in the collaboration between interaction (Ix) designers and agile developers. They found that sketches and stories support creation and reflection, facilitate resolution of contradiction, and work at a level of consciousness that is below the level of self-awareness.

At a higher level of task abstraction, Barksdale and others [6,7] presented a method for connecting the various domains through the use of concept mapping. They argue that collaborative concept mapping can alleviate politics on agile usability teams and can improve their interaction by facilitating communication while enabling role autonomy. In their approach, usability experts add scenarios to the map, developers
link stories to the scenarios, and both collaborate on that link to provide deeper rationale for the association between the scenarios and stories. They found that—although there is a need for improvement in agile usability and a concept mapping approach is promising for improving team interaction in agile usability environments—role familiarity is important, collaboration and communication don’t imply one another, and there is value in providing steps for sharing knowledge versus just the structure to do so.

3.7. Chapter Summary

Toward bringing the user closer to the development process, various approaches have been employed to bridge the gap between usability and software development. These have ranged in level of effort from the superficial selection of practices to the more complex infrastructural changes of influencing how knowledge is constructed. The findings from the literature review suggest that there’s a need for greater emphasis on two key aspects of agile usability teams: 1) understanding the effects of social capital within agile usability teams; and 2) understanding how agile usability teams are governed.

In the remainder of this work, lessons from these findings are studied. Namely, the focus is on understanding the role of social capital and network governance on user experience knowledge management (UKM). UKM was selected as the outcome given that one goal of agile user experience is to improve the user experience of software. One factor influencing accomplishment of this goal is the management of user experience knowledge. The expectation is that as the social capital and governance structure change, so does the way user experience knowledge is managed. What follows are the presentation and explanation of this theoretical framework and details about the proposed study.
Chapter 4: Team Interaction Framework

Chapter Outline:

Social versus Cognitive Theories
Social Capital
Network Governance
Knowledge Management
Agile User Experience Team Interaction Framework
Chapter Summary

After reading this chapter, you should:

● Understand the purpose for using primarily social theories over cognitive theories
● Understand each theoretical component of the Team Interaction Framework
● Understand how the various theories are related to accomplish the research objectives
Investigating the interplay between social capital, network governance, and user experience knowledge management provides a lens through which to analyze and understand the influence of social capital and network governance in agile user experience software teams. This section presents the necessary background to understand the Team Interaction Framework and explains how it is used in this research.

4.1. Social versus Cognitive Theories

Cognitive methods are commonly used to analyze social environments in human computer interaction (HCI). Cognitive frameworks such as Activity Theory [88], Situated Action [114], and Distributed Cognition [59,60] are examples of methods that have been employed in the design of human-centered technology [89,115]. Activity Theory, the oldest and most developed approach, uses an activity as the unit of analysis to understand “the concept of artefact-mediated and object-oriented action” [89,115]. Situated Action, using the relationship between individuals and their environment as the unit of analysis [89,114], focuses on exploring the circumstances in which this relationship occurs [115]. Distributed cognition utilizes a distributed cognitive system of actors and artifacts as the unit of analysis to understand how individual knowledge is created, represented, transformed and propagated among groups of individuals in an environment [89,115].

However, such cognitive methods do not sufficiently facilitate an understanding of the social relationships and governance structure among team members. Actor relations are only considered relative to their environment and the artifacts they create. Social network theories acknowledge that knowledge is a resource that can influence social relationships, but analytical emphasis is on how knowledge is influenced by social relationships, relational attributes within the team, and the structure and communication exchanges between people [87]. Traditional hierarchical associations between and among teams (and their management) are less flexible and can make it challenging for teams to adjust to external forces often experienced on software projects [85]. By viewing software teams collectively as a network of informal social structures, there may be opportunity to improve how user experience knowledge is managed. Hence, the unit of analysis, and the primary interest in this work, is the social network—the network of communication links among individuals—and how such networks are governed.

4.2. Social Capital

Social relationships play a key role in how team members communicate. They can make it easier for team members to approach one another, serve as a support system during difficult times, and provide a more
fulfilling work experience. A team that dismisses their social dynamics and structure might struggle in accomplishing their tasks, whereas those that leverage these characteristics might function more fluidly through awareness of influential social factors. For example, brainstorming with other team members may help a member resolve some of the issues experienced with a colleague that prevents them from focusing on the project. Talking with other members about the experience may also help establish rapport and build trust within the team.

Social capital provides a means for capturing and understanding these social dynamics. It has been defined in many ways [3] and used across various disciplines—such as political science, public administration, sociology, and management (Table 2). However, as is evident in the table, there is still a lack of consensus on what it entails and its purpose. For some, social capital is merely a reconstruction of social network theory, and thus an unnecessary construct. For others, it provides a means for providing a more accessible measure of such dynamics and presumes the notion that social networks have value. The most fitting, and selected, definition for this work is that provided by Nahapiet and Goshal [87].

Table 2: Definitions of Social Capital

<table>
<thead>
<tr>
<th>Authors</th>
<th>Domain</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hanifan 1916</td>
<td>Political Science</td>
<td>“...that in life which tends to make these tangible substances count for most in the daily lives of a people, namely, good-will, fellowship, mutual sympathy and social intercourse among a group of individuals and families who make up a social unit, the rural community, whose logical center is the school.”</td>
</tr>
<tr>
<td>Bourdieu 1986</td>
<td>Sociology</td>
<td>“The aggregate of the actual or potential resources which are linked to possession of a durable network of more or less institutionalized relationships of mutual acquaintance or recognition.”</td>
</tr>
<tr>
<td>Author</td>
<td>Field</td>
<td>Quote</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Schiff 1992</td>
<td>Management</td>
<td>&quot;...the set of elements of the social structure that affects relations among people and are inputs or arguments of the production and/or utility function.”</td>
</tr>
<tr>
<td>Burt 1992</td>
<td>Sociology</td>
<td>&quot;...friends, colleagues, and more general contacts through whom you receive opportunities to use your financial and human capital.”</td>
</tr>
<tr>
<td>Fukuyama 1997</td>
<td>Political Science</td>
<td>&quot;...the existence of a certain set of informal values or norms shared among members of a group that permit cooperation among them.”</td>
</tr>
<tr>
<td>Nahapiet &amp; Goshal 1998</td>
<td>Management</td>
<td>&quot;...the sum of the actual and potential resources embedded within, available through, and derived from the network of relationships possessed by an individual or social unit.”</td>
</tr>
<tr>
<td>Putnam 2000</td>
<td>Political Science</td>
<td>&quot;...connections among individual-social networks and the norms of reciprocity and trustworthiness that arise from them.”</td>
</tr>
</tbody>
</table>

Social capital is commonly assessed using social network analysis (SNA). It is performed by measuring and analyzing the properties of social networks [31]. These measures include, for example, centrality (can be used to assess how connected an individual is as a central actor), cohesion (the degree to which members are directly tied to each other, which can be used to identify cliques), and density (the amount of connectivity among network members and an indicator of cohesion). Such data is useful for gaining insight about how valuable the network is, how well social capital is leveraged in a network, and how it can be adjusted for desired outcomes.

### 4.3. Network Governance

Network governance theory emphasizes the coordination of informal social systems through establishing or leveraging structures to guide network activities and network-level outcomes (e.g., network efficiency) [61,100]. This means changing how the network functions to achieve a desired result. It is different from the structural dimension of social capital because it is concerned with the structure of decision-making in the network instead of merely the positioning of members in the network.
The network governance theory used in this study was developed by Provan and Kenis [100]. They define network governance as involving “the use of institutions and structures of authority and collaboration to allocate resources and to coordinate and control joint action across the network as a whole.” This theory is used because it not only distinguishes between organizational and network governance (with a focus on the network), but also because it is flexible—able to accommodate application in a variety of domains. Consistent with the intentions of this work, they view the network “as a variable, examining different network governance configurations and the conditions for the effectiveness of each form.”

Provan and Kenis’ network governance theory posits that:

**Proposition 1:** The greater the inconsistency between critical contingency factors (i.e., trust, size, goal consensus, and the nature of the task) and a particular governance form, the less likely that particular form will be effective, leading either to overall network ineffectiveness, dissolution, or change in governance form.

**Proposition 2:** Shared network governance will be most effective for achieving network-level outcomes when trust is widely shared among network participants (high-density, decentralized trust), when there are relatively few network participants, when network-level goal consensus is high, and when the need for network-level competencies is low.

**Proposition 3:** Lead organization network governance will be most effective for achieving network-level outcomes when trust is narrowly shared among network participants (low-density, highly centralized trust), when there are a relatively moderate number of network participants, when network-level goal consensus is moderately low, and when the need for network-level competencies is moderate.

**Proposition 4:** Network Administrative Organization (NAO) network governance will be most effective for achieving network-level outcomes when trust is moderately to widely shared among network participants (moderate density trust), when there are a moderate number to many network participants, when network-level goal consensus is moderately high, and when need for network-level competencies is high.
**Proposition 5:** Networks face a **tension between the need for administrative efficiency and inclusive decision making.** In shared-governance networks, the tension will favor inclusion; in lead organization-governed networks, the tension will favor efficiency; and in NAO-governed networks, the tension will be more balanced but favor efficiency.

**Proposition 6:** Networks face a **tension between the need for internal and external legitimacy.** In shared-governance networks, the tension will favor internal legitimacy; in lead organization-governed networks, the tension will favor external legitimacy; and in NAO-governed networks, both sides of the tension will be addressed but in a sequential fashion.

**Proposition 7:** Networks face a **tension between the need for flexibility and the need for stability.** In shared-governance networks, the tension will favor flexibility; in NAO- and lead organization-governed networks, the tension will favor stability.

**Proposition 8:** Assuming network survival over time, as network governance changes, it is likely to **evolve** in a predictable pattern from shared governance to a more brokered form and from participant governed to externally (NAO) governed. Evolution from shared governance to either brokered form is significantly more likely than evolution from a brokered form to shared governance. Once established, evolution from an NAO to another form is unlikely (i.e., inertia is strongest when the governance form is more formalized).

This theory holds that there is an expectation that the independent variables (viz., network structural, cognitive, and relational critical contingency factors) influence the dependent variables (viz., desired team user experience knowledge management) because the configuration of social capital factors in a social network affect network-level outcome (viz., user experience knowledge management).

These propositions align with the aims of this study and substantiating them at the team level, where possible, may prove beneficial to gaining additional perspectives about agile user experience software teams. Hence, network governance form is part of the theoretical framework like that of a statistical moderator variable, where the interaction between the level of social capital and changes in user experience knowledge management are explained by the form(s) of governance in the team.
4.4. Knowledge Management

The way knowledge is created and shared has important implications and consequences in work team collaboration. What, how, and when knowledge is acquired and exchanged can influence the team’s effectiveness and efficiency. If team members do not have a clear understanding of the knowledge they are sharing or attach different meanings to knowledge that is gained, then collaboration can become strained and ineffective. However, if knowledge can be managed in a way that enables a shared understanding and is exchanged at opportune moments, then it could put team collaboration on a more solid footing—possibly increasing the likelihood of the team realizing their objectives.

4.5. Agile User Experience Team Interaction Framework

Framework Goals and Rationale

The goal of the Team Interaction Framework (Figure 2) is to fulfill the aforementioned purpose of this research. Namely, it helps with the first analytical purpose of providing researchers and practitioners with an understanding of how socio-cognitive factors and network governance forms influence the integration of software developers and user experience professionals in agile environments (i.e., analytical purpose). Data collection, analysis, and interpretation is informed and guided by the modeling of relevant concepts and theories. It also aids with the second purpose of providing guidance to practitioners on ways to adjust how user experience knowledge is managed in their agile user experience teams (i.e., practical purpose). In essence, the understanding gained from analysis and interpretation from the first purpose is used to expand the framework into a tool practitioners can use to guide the interactions within their team and change how user experience knowledge is managed.

![Figure 2: Team Interaction Framework](image)

How Social Capital Fits into the Framework
Social capital is incorporated into the framework as a condition that helps to explain the influence social capital has on how user experience knowledge is managed. Its inclusion has import for both analytical and practical purposes. From an analytical perspective, its inclusion in the framework facilitates capturing data about the team’s social capital. Its inclusion also helps to answer the question of what and how best to measure social capital. From the practical perspective, it helps give practitioners insight about what targeted changes they can make to their social capital to create change likely to achieve the desired effectiveness for how user experience knowledge is managed.

How Network Governance Fits into the Framework
Network governance serves as another condition (albeit, a moderator-like variable) that helps to explain how the relationship between social capital and user experience knowledge management is affected by the governance form of the network (or team). Considering network governance as a moderator-like variable means that some governance forms may influence the relationship between social capital and user experience knowledge management and some may not.

How User Experience Knowledge Management Fits into the Framework
User experience knowledge management is the outcome of interest in the framework and of this study. How user experience knowledge is managed has implications for the resulting user experience of the software product, team management, and customer satisfaction. For example, ineffective or inefficient management of user experience knowledge could result in user experience decisions not being incorporated into the software because of a breakdown anywhere between the time it was created and its potential use. Although all user experience knowledge may not be utilized, there is value in knowing where and why it lost traction and whether it was an intentional or unintentional decision to not use that knowledge.

4.6. Chapter Summary
The Agile User Experience Team Interaction Framework is premised on the idea that social and cognitive dynamics can encourage or undermine the effectiveness of each other, and that both are collectively required to accomplish the goals of agile user experience software teams. They work together such that the social relationships strengthen the communication medium through which knowledge flows. In essence, the social relationship is the communication channel through which knowledge passes to
accomplish some shared objective. The following chapter specifies the plan for how the study will be conducted that will provide the data and enable its analysis consistent with this theoretical framework.
Chapter 5: Research Design and Methods

Chapter Outline:

- QCA Overview and Rationale
- Research Questions
- Propositions
- Methods and Procedures
- Chapter Summary

After reading this chapter, you should:

- Understand which components and methods will be used to conduct the study
- Understand Qualitative Comparative Analysis (QCA) and its role in this research
- Understand Social Network Analysis (SNA) and its role in this research
This chapter details how the study was conducted to obtain an empirical perspective of the theoretical framework and to facilitate what Ragin calls an iterative dialogue between ideas (i.e., theory) and evidence (i.e., cases)—a progressive exchange between condition selection in QCA and the cases toward data interpretation [108]. Following is an overview of QCA, the research questions and propositions of this study, and a presentation of the research design and methods used in this study.

5.1. QCA Overview and Rationale

Qualitative comparative analysis (QCA) was developed by Charles Ragin [103,104,105,107,108] to fill the need for a method situated between statistical (i.e., quantitative) and in-depth case-oriented (i.e., qualitative) methods. It is commonly used in the social sciences to study entities typically limited in quantity such as political nation-states. It is best suited for studies that have a small to intermediate N—in essence, studies too small for statistical analysis but too large for a traditional comparative case analysis. In QCA, the conditions and the outcome comprise a configuration. The table below (Table 3) highlights the key differences between statistical analysis, qualitative comparative analysis, and case analysis.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Statistical Analysis</th>
<th>QCA</th>
<th>Case Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical Foundation</td>
<td>Probability Theory</td>
<td>Set Theory</td>
<td>Mills Methods</td>
</tr>
<tr>
<td>Algebraic Foundation</td>
<td>Linear Algebra</td>
<td>Boolean &amp; Fuzzy Algebra</td>
<td>N/A</td>
</tr>
<tr>
<td>Data Type</td>
<td>Variable-oriented</td>
<td>Variable (in context of a case)</td>
<td>Case</td>
</tr>
<tr>
<td>Distribution Space (n)</td>
<td>Large N (&gt;100)</td>
<td>Small to intermediate N</td>
<td>Small N (&lt;5)</td>
</tr>
<tr>
<td>Unit of Analysis</td>
<td>Sample</td>
<td>Case</td>
<td>Case</td>
</tr>
<tr>
<td>Variable Influence</td>
<td>Isolation</td>
<td>Conjunctural</td>
<td>N/A</td>
</tr>
<tr>
<td>Explanatory Variable</td>
<td>Independent</td>
<td>Conditions</td>
<td>N/A</td>
</tr>
<tr>
<td>Dependent Variable</td>
<td>Response</td>
<td>Outcome</td>
<td>N/A</td>
</tr>
</tbody>
</table>
At first glance, the differences between the methods might appear as a matter of semantics. A more thorough assessment, however, reveals some of the nuances. For example, what constitutes a sufficient sample size is grounded in the underlying algebraic method used. Linear algebra requires sample data that sufficiently represents the population to make valid inferences. This is different from Boolean algebra, where emphasis is on the logical space and the number of cases that support or refute a theory. It is important to note that cases are not samples of a population, but instances of support or opposition of a theory [125]. This distinction about population plays a key role in the difference in the level of inference one can make, and equating a sample to a case can yield misleading conclusions. More information about QCA and fuzzy sets is available in Appendix D (the QCA Primer) and in Ragins’ book on the use of fuzzy sets in social science [107].

5.2. Research Questions

Previous studies, as presented earlier, have largely taken a cognitive- or engineering-centered approach to agile user experience integration. This research aims to complement the literature on cognitive- or engineering-centered approach through study of the social interaction in agile user experience teams. Hence, this research seeks to address the following question(s):

1. How do social capital factors and network governance forms influence how user experience knowledge is managed in agile user experience software teams?
   
a. [current descriptive existence] What social network governance forms are currently used, what are the current levels of social capital, and how is user experience knowledge currently managed in agile user experience software teams?

b. [causal relationships] What influence do existing social capital factors and network governance forms have on current user experience knowledge management in agile user experience software teams?
c. [optimal relationships] What combinations of social capital factors and social network
governance forms help agile user experience software teams to achieve their desired level
of user experience knowledge management?

5.3. Propositions

Propositions are stated differently in QCA than they are in a statistical design [122]. In QCA they are stated
as a conditional (i.e., if-then) statements, necessary and/or sufficient assertions, or as logical implications.
The propositions in this study are assessed according to set membership. Moderation, shown as the down
arrow on the left side of the figure, is a statistical technique that seeks to explain to what extent the
relationship between two variables is dependent on, or influenced by, a third variable. Moderation,
however, is not a concept associated with QCA, but is inherently assessed given the configurational nature
of QCA. It is important to note that the use of propositions (versus hypotheses) is intentional. The purpose
of this research is to build a theoretical framework versus test the framework.

Following are the propositions for this study based on the research questions:

RQ1(a): It is anticipated, in a given project team and within a release, that all three governance
forms (viz., the shared, lead, and network administrative) will be found, that social capital levels
will vary at various stages of development, and that the level of user experience knowledge
management will vary based on the governance form, the level of social capital, and the stage of
development.

RQ1(b): It is anticipated that all combinations of social capital, governance form, and user
experience knowledge management are employed at different stages of development within a
release cycle.

RQ1(c): It is anticipated that an optimal configuration for achieving the desired user experience
knowledge management for teams in the beginning of a release is medium social capital and a
shared governance form given the need to determine and agree on scope. It is anticipated that
an optimal configuration for achieving the desired user experience knowledge management for
teams in the middle of a release is low social capital and lead team governance given the need for
focus on implementation. It is anticipated that an optimal configuration for achieving the desired user experience knowledge management for teams at the end of a release is high social capital and network administrative organization given the need for quality assurance before deployment.

5.4. Methods and Procedures

Comparative Case Study Design

Comparative case study research design is concerned with the identification and analysis of cases toward making analytic (or theoretical), versus statistical, generalization [125]. This means that each case is like an experiment instead of part of a sample. Whereas the emphasis in statistics is on generalizing to a population, the emphasis in analytic generalization is on whether the results of each case support or refute a theory [125]. Although comparative case study research is often complemented with statistical analysis, this work uses an inherent mixed methods approach—Qualitative Comparative Analysis (QCA)—to analyze cross-case results and make inferences.

Study Phases

This research is organized into three phases: the development, case study, and substantiation phases (Table 4).

<table>
<thead>
<tr>
<th>Study Phase</th>
<th>Purpose</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>To create and validate study materials and instruments</td>
<td>Subject Matter Experts</td>
</tr>
<tr>
<td>Case Study</td>
<td>Study specific cases/teams</td>
<td>Practitioners</td>
</tr>
<tr>
<td>Expert Review</td>
<td>To obtain expert feedback on the framework</td>
<td>Subject Matter Experts</td>
</tr>
</tbody>
</table>

Case Selection

Each case study entails the assessment of a team’s social capital, network governance form, and how user experience knowledge is managed. The unit of analysis for each case is the interaction between software developers and user experience professionals. Hence, these assessments will capture the interaction between software developers and user experience professionals within the network boundary of a
software product or project team. Although there is import for understanding the interaction among all team members, the implications and consequences for software developers and user experience professionals are of primary concern for this research given the direct connection between system development and the user experience of the product.

Condition Selection

Condition (or variable) selection should be informed by theoretical or substantive knowledge, but should also be kept to a critical few to reduce the risk of limited diversity in QCA [108]. Limited diversity occurs when the cases only cover a small portion of the total possible logical combinations of conditions (i.e., the logical or property space). In this study, measures of the social capital dimensions and the network governance form serve as the conditions. The phases of user experience knowledge management serve as the outcomes. A detailed description of the conditions and outcomes, and the measures used to collect data on each are provided later in the chapter.

Conditions and Outcome

Following is an overview of the key conditions that were used in this study.

Table 5: Summary of the conditions, definitions, and measures of each condition

<table>
<thead>
<tr>
<th>Outcome / Conditions</th>
<th>Operational Definition</th>
<th>Measures</th>
</tr>
</thead>
</table>
| User Experience Knowledge Management (Outcome) | User experience knowledge management is defined as the monitoring or influencing of the progression of user experience knowledge through the stages of construction, embodiment, dissemination, and use. | UKC: Level of user experience ideas discussed  
UKE: Level of user experience ideas embodied into artifacts  
UKD: Level of user experiences ideas made accessible to the team  
UKU: Level of user experiences ideas implemented into a build |
| Social Capital (Condition) | Social capital is defined as being comprised of structural, cognitive and relational | SD: Network density and centralization  
CD: Level of agreement on shared meaning  
RD: Level of agreement on trust (Jarvenpaa & Leidner 1999); level of statistical agreement |
Network Governance (Condition)

<table>
<thead>
<tr>
<th></th>
<th>Network governance is defined as the form by which a network is governed (based on who makes network-related decisions). Provan identifies three forms of governance: shared (all members), lead (one or a few), and network administrative (external).</th>
<th>Level of agreement on governance form by participants measured via questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimensions</td>
<td>of norms and values using Krippendorff’s alpha</td>
<td></td>
</tr>
</tbody>
</table>

Outcome

**User Experience Knowledge Management (UKM).** Various philosophies underlie the meaning of knowledge, types of knowledge (e.g., tacit/explicit, practical/theoretical, declarative/procedural/causal), and how to manage knowledge (e.g., as intellectual capital [73,74,87]). Ackoff defines knowledge as the application of data and information [1]. Although information may contain relationships between data, it is not necessarily wholly organized as a body of work.

The McAdam and McCreedy knowledge management model [73,74]—derived from Demarest and scaled from the organizational to the team level of analysis—is used as the basis for understanding user experience knowledge management in this research. Foundational to this model is the paradigm that knowledge is socially constructed [73,74]. The model—originally created by Clark and Staunton [25], later adapted by Demarest [32], and further modified by McAdam and McCreedy to acknowledge the dual paradigms [73,74]—has the following four phases, which serve as the operational definition of user experience knowledge management. The words “user experience” were added to the name of each knowledge management phase to indicate the kind of knowledge of particular interest in this research.

- **User Experience Knowledge Construction (UKC).** Knowledge construction (and acquisition) is “the process of discovering or structuring a kind of knowledge” [32]. Construction was measured, via a seven point likert response scale of frequency, as member perceptions of user experience
ideas discussed within the team. This phase includes the construction of new knowledge and the acquisition of existing knowledge. For example, user experience professionals may construct user experience knowledge by interviewing users to generate ideas on how to improve the user experience of a system.

- **User Experience Knowledge Embodiment (UKE).** Knowledge embodiment is “the process of choosing a container for knowledge once it is constructed (e.g., a design artifact)” [32]. Embodiment was measured, via a seven point likert response scale of frequency, as member perceptions of user experience ideas that are incorporated into a tangible [design] artifact (e.g., wireframe). From the interviews in the previous example, each user experience idea included in the UI prototype is embodied knowledge.

- **User Experience Knowledge Dissemination (UKD).** Knowledge dissemination is “the human processes and technical infrastructure that make embodied knowledge available within the firm who use the documents and the bodies of knowledge those documents contain” [32]. Dissemination was measured, via a seven point likert response scale of frequency, as member perceptions of user experience ideas that have been shared with the remaining software project team members. Continuing with the example, when the UI prototype is shared with the remainder of the software product or project team it is considered disseminated knowledge.

- **User Experience Knowledge Use (UKU).** Knowledge use is “the production of commercial value for the customer” [32]. Use was measured, via a seven point likert response scale of frequency, as member perceptions of user experience ideas that have been incorporated into a software product build. The software product (and its features by implication) provides the most commercial value for the customer. Concluding the example, the elements from the UI prototype included in the build is used knowledge.

**Conditions**

**Social Capital (SC).** Social capital is an application of social network theory and is conceptually defined as “the sum of the actual and potential resources embedded within, available through, and derived from the network of relationships possessed by an individual or social unit” [87]. Value is associated with social capital similar to other forms of capital such as human capital and financial capital. Nahapiet and Goshal operationally define social capital as being comprised of three dimensions – a structural, cognitive, and
relational dimension [87]. Following is the definition of these dimensions in the context of how they were used in this study.

- **Structural Dimension (SD).** Social networks are thought of as having static and dynamic properties. Of these two, the structural dimension addresses the static property. Specifically, it is concerned with the structure of, and connections in, the network - the impersonal characteristics [87]. Measures of the structural dimension were network density and centralization [87]. A social network analysis tool was used to obtain the level of density and centralization in the network [19].

- **Cognitive Dimension (CD).** The cognitive dimension of social networks is concerned with the resources that facilitate knowledge exchange within the network such as shared representations, interpretations, and systems of meaning [20,87]. It is also associated with whether learning occurs within the network. Shared cognition was assessed as the amount of learning that occurs within the network, which was measured by team member knowledge of domains other than their own on the software team. Namely, software developer knowledge of user experience (and vice versa) was assessed. A fuzzy score was used to denote their level of knowledge based on responses to likert questions.

- **Relational Dimension (RD).** The relational dimension describes the dynamic property of social networks. It is concerned with the personal relationships between actors in a network. The relational dimension was assessed by measuring the level of trust and agreement among members on norms and values. A fuzzy score was used to denote the average level of trust within the network and the amount of agreement on the norms and values.

**Network Governance Form (NGF).** Provan and Kenis present three inter-organizational network governance forms - shared, lead organization, and network administrative organization - and the related critical contingency factors that are predictors of each form [100]. However, network governance is considered at the inter-team (user experience and software developers) instead of the inter-organizational level in this research. Thus, network governance is operationalized as follows:
• **Shared Governance (SG).** The shared network governance form is the most decentralized form where all members of the network govern how the network functions. One example of this form is if all members from both the user experience and software teams share in making decisions about how user experience knowledge will be administered.

• **Lead Governance (LG).** The lead team governance form exists when one or a few members govern how the network functions. An example of this structure could be if the user experience team (or a subset of it) makes decisions about how user experience knowledge will be managed for the software team.

• **Network Administrative Governance (NAG).** The network administrative governance form exists when an independent (viz., external) team governs how the network functions. This might occur, for example, if a management team (e.g., a project management office) makes decisions about how user experience knowledge will be managed.

*Validity and Reliability*

Validity of case studies is commonly questioned compared to statistical methods given the subjectivity involved. Several statistical tests exist to assess validity in research - construct validity, internal validity, external validity, and reliability. Although not statistical tests, these forms of validity and reliability are also applicable to case study research [125]. Hence, validity and reliability are explicitly addressed in this section.

**Construct Validity**

Construct validity is concerned with ensuring that the proper constructs (or variables) are employed to allow one to draw valid conclusions. It entails ensuring variables are appropriately operationalized and measured. Construct validity can be established in case study research through the use of multiple sources of evidence and maintaining a chain of evidence from research questions to conclusions [125]. This study mitigates construct validity concerns by (1) using existing constructs to measure respective concepts that have been used in the literature and using multiple sources of evidence, and (2) maintaining a chain of evidence, which provides explicit links connecting the research questions to conclusions drawn (Figure 3).
Internal Validity

Internal validity is concerned with the causality among conditions and outcomes. This can be achieved through techniques such as using logic models, explanation building, and addressing rival explanations [125]. This study attempts to provide explanations through making logical inferences through QCA. It also affords building explanations since conditions serve as explanatory factors, and considers rival explanations via inclusion and elimination of logical remainders. It’s important to note that using QCA was a strategic decision due to the sample size required to make statistical inference.

External Validity

An important aspect of scientific studies is whether the results are generalizable. This study seeks to achieve external validity but there are subtle differences about what this means since it uses a comparative case study design versus a statistical design. Statistical designs make generalizations about a population via statistical inference from data collected about a sample of that population. This is what Yin considers Level One inference [125]. This is distinct from case study research because cases are viewed not as samples but experiments [125]. Case study research is more concerned with Level Two inference (i.e., analytic generalization) – in essence, whether case results support a broader theory of interest or a rival theory [125]. For example, when a case supports a given theory and not its rival theory, there is greater analytic generalizability [125].

External validity beyond an immediate case study is achievable through the use of theory and replication [125]. This is accomplished through the use of a theoretical framework linking social capital theory, network governance theory, and knowledge management theory. Replication is incorporated into this research design by the development of multiple cases. Some QCA configurations (or causal explanations) are likely to have multiple cases associated with them. QCA facilitates analytic generalization via the comparison of multiple cases and identification of logical remainders that can be used for the analysis of counterfactual cases (viz., a configuration that lacks empirical instances) or rival theories (i.e.,...
contradictory configurations). Analytic generalization is also possible in QCA by reducing the limited diversity so cases are not merely described, and increasing the amount of evidence (i.e., number of cases) for configurations [108].

Reliability
Case study reliability can be increased through the use of a case study protocol, which facilitates repeatability [125]. The protocol should detail what data will be collected for each case such as its purpose, relevant questions, data collection procedures, an outline of the case report, and other material necessary for another researcher to repeat the study. The research proposal served as the protocol. Additionally, following the procedures of QCA enhances analytical repeatability.

**Design Benefits and Limitations**
The comparative case study design, in concert with QCA, has the unique benefits of: (1) providing a formal approach to solving the otherwise difficult problem of ensuring rigor, and (2) facilitating an understanding of causality as a combination of conditions versus attempting to isolate a single factor. Whereas statistics aims to isolate factors to determine which one is most correlated with some response (i.e., identifying the net effect), QCA enables identification of which factors, collectively, help to explain the outcome—called causal complexity [104]. This is important given that, in practice, it is rare that there is one single factor that explains an outcome, and more importantly, there is greater actionable insight in understanding which combination of conditions yields a (un)desired outcome. Also, attempting to measure team interaction statistically would lead to results with little significance in this study because of the small sample size. However, qualitative studies still entail rigor. The design used in this study facilitates this formal structure and rigor.

The limitations of the comparative case study design are the inherent limitations of Boolean versus linear algebra, deterministic versus probabilistic causation, and population versus analytical generalization. In essence, there are fewer operations available in Boolean than linear algebra, the asymmetry of deterministic versus the symmetry of probabilistic causation, and the inability to generalize to a population. These individual limitations combined and considered in the context of the overall design, however, were strengths for this research. The intent of this research was to develop a theoretical understanding of team interaction in agile user experience software teams, which could later be tested across populations using statistical means as future research.
Participants
Study participants included practitioner software developers and user experience professionals that were, or had been within the past year, members of a multidisciplinary agile user experience software team. Computer science and other software and user experience subject matter experts were solicited to provide expert feedback on the survey and for substantiation. During the case study phase, participants had to be an active member of an agile user experience team, and they must have remained on the software team for the duration of the study.

Selection Strategy
The convenience method was used to select participants. Hence, participants were selected based on their accessibility and relevance to the study. This method was chosen given the difficulty of gaining access to industry practitioners.

Materials
Participants were provided a description of the study. A Team Interaction Assessment was developed and administered to assess the social capital in the team, assess the network governance form used in the team, assess how user experience knowledge was managed in the team, and to collect background data about members of the team. Four subject matter experts (viz., graduate students and practitioners) were asked to provide initial impressions after reviewing the instruments (most closely related to face validity [70]). This content was then assessed to identify which changes might help ease the administration of the survey and, thereby, the accuracy of participant responses during data collection.

Data Collection Procedures
Data collection entailed a variety of case study methods. Approximately 20 person-hours per week were dedicated to each team. Direct observation, interviews, and questionnaires were used to understand the team’s interaction, how user experience knowledge was managed, and the context in which the interaction occurs. Due to the industrial nature of this research, data collection was bound by the constraints imposed by company policies and management. These constraints include, for example, the limited time investment available to collect data, limited availability of proprietary information, and required anonymity of the company, its employees, and its customers. These constraints accommodated by keeping the amount of time required of participants as low as possible and removing any identifying information from the data used about the company and its stakeholders.
Data Analysis
Data analyses occurred using a mixed methods approach. Social network analysis (SNA) was used to analyze the structural dimension of social capital. Qualitative data collected via observation and survey were analyzed using thematic analysis to identify patterns in support of, or counter to, findings from other analyses. Data across cases were analyzed using qualitative comparative analysis (QCA) to make inferences about the relationship between the conditions and the outcomes of the theoretical framework.

Social Network Analysis
Social network analysis (SNA) is an analytical technique used for understanding the communication links and dynamics within a social network. The primary diagram created during SNA is a sociogram that graphically represents the relationships as nodes (i.e., actors) and edges (i.e., links). Various properties of a social network provide insight about the type of interaction in the network. Common properties include the presence or absence of structural holes and centrality as an indicator of control and power distribution in the network [47], cohesion as an indicator of cohesive bonds, and density as an indicator of the complexity in the network. The benefit of SNA is its simplicity in understanding social networks via sociograms. A drawback is that the results are not sufficient for understanding the causal relations within a network and are only as good as the survey questions and responses. Social network analysis was performed using software called ORA [19].

Qualitative Comparative Analysis
Analysis Procedures
Fuzzy-set QCA will be used to evaluate causal relations between the social capital and network governance conditions and the outcome of how the team manages user experience knowledge. fsQCA will be performed in the following manner:

1. Responses and observational data will be aggregated and mapped to fuzzy-set values, relevant thresholds will be determined, and the data will be calibrated [108].

2. The test of necessity will be performed on configurations and the plots will be examined to eliminate any configurations prior to conducting the standard analysis test of sufficiency.

3. The standard analysis will be conducted. At this stage, the truth table is generated and thresholds are executed to determine which cases will be included in the standard analysis.
4. The analysis concludes with three kinds of models that suggest an explanation of the outcome based on the evidence. These three types of models provide a complex, intermediate, and parsimonious solutions. The complex solution will be used given the emphasis on empirical results in the solution.

5.5. Chapter Summary

This chapter detailed the research questions, propositions, research design and methods that work in concert to provide an understanding about the relationship between social capital, network governance, and user experience knowledge management. Social network analysis is used to explicate the relationships between software developers and user experience professionals in agile user experience software teams. The comparative case design and qualitative comparative analysis are used to identify relationships between the theoretical components.
Chapter 6: Results and Discussion

Chapter Outline:

Summary of Case Results
Cross-Case Results
Evaluation of Propositions
Framework Expert Review
Chapter Summary

After reading this chapter, you should:

- Have a general understanding of the case study findings
- Understand to what extent the results addressed the propositions
- Understand the results from cross-case analysis via QCA
This chapter provides the results from across the case studies based on the qualitative comparative analysis and interpretations in light of the research questions. A summary of the key findings from each case study are included in this chapter, but the results from each case study are in the appendix.

6.1. Summary of Case Results

This section contains the summary of key findings from the analyses of each case study as context for the cross-case findings. Team A (viz., case A) was situated in a small-sized information technology (IT) services government contracting company. The team was comprised of a Product Owner, Testers, Developers, UX professionals, Documentation staff, and Community Support staff. The project environment was agile—iterating on features prior to their collective release.

Team B (viz., case B) was situated in a mid-sized software services development company. The team was comprised of a Product Manager, Technical Product Managers, Quality Engineers, Developers, and UX professionals. The project environment was lean—utilizing continuous integration and deployment.

Quantitative Summary

The results for the structural dimension of social capital were at the team level (i.e., including all team members that participated in the study). The remaining results reflect the interaction between the software developers and user experience professionals on the team.

Table 6: Case summary results

<table>
<thead>
<tr>
<th>Measure</th>
<th>Method</th>
<th>Case A</th>
<th>Case B</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC.Structural.Collaboration.Density</td>
<td>SNA</td>
<td>34.6%</td>
<td>40%</td>
</tr>
<tr>
<td>SC.Structural.Collaboration.Centralization</td>
<td>SNA</td>
<td>30.5%</td>
<td>26.6%</td>
</tr>
<tr>
<td>SC.Structural.Help.Density</td>
<td>SNA</td>
<td>35.9%</td>
<td>40%</td>
</tr>
<tr>
<td>SC.Structural.Help.Centralization</td>
<td>SNA</td>
<td>21.3%</td>
<td>27.7%</td>
</tr>
<tr>
<td>SC.Structural.Advice.Density</td>
<td>SNA</td>
<td>34.6%</td>
<td>36.1%</td>
</tr>
<tr>
<td>SC.Structural.Advice.Centralization</td>
<td>SNA</td>
<td>30.5%</td>
<td>33.7%</td>
</tr>
<tr>
<td>SC.Cognitive.SharedMeaning</td>
<td>Likert mean</td>
<td>5.75 (out of 7)</td>
<td>6.36 (out of 7)</td>
</tr>
<tr>
<td>SC.Relational.Trust</td>
<td>Likert mean</td>
<td>5.92 (out of 7)</td>
<td>6.12 (out of 7)</td>
</tr>
<tr>
<td>SC.Relational.Norms</td>
<td>Krippendorff's alpha</td>
<td>.125</td>
<td>-.38</td>
</tr>
<tr>
<td>SC.Relational.Values</td>
<td>Krippendorff's alpha</td>
<td>.727</td>
<td>.097</td>
</tr>
<tr>
<td>Network Governance</td>
<td>Agreement (custom)</td>
<td>50%</td>
<td>49%</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>KM.Construction</td>
<td>Likert mean</td>
<td>5 (out of 7)</td>
<td>4.1 (out of 7)</td>
</tr>
<tr>
<td>KM.Embodiment</td>
<td>Likert mean</td>
<td>4.5 (out of 7)</td>
<td>5.2 (out of 7)</td>
</tr>
<tr>
<td>KM.Dissemination</td>
<td>Likert mean</td>
<td>4 (out of 7)</td>
<td>4.8 (out of 7)</td>
</tr>
<tr>
<td>KM.Use</td>
<td>Likert mean</td>
<td>6.5 (out of 7)</td>
<td>4.8 (out of 7)</td>
</tr>
</tbody>
</table>

**Thematic Results Summary**

**Conflict between UX and Dev.** This theme reflects conflict, or its potential, between user experience professionals and developers. The nature of the conflict was different across cases, most notably in terms of geo-location. Whereas Team A experienced conflict as a result of synchronization, Team B experienced conflict resulting from physical distance.

**Interaction Improvement Strategies.** This theme reflects strategies that have been employed toward minimizing or resolving conflict. Interaction improvement strategies also differed across cases. Team A employed more socially oriented strategies (i.e., brownbags, providing positive feedback). Team B employed a structural change (i.e., addition of a TPM), which decreased the direct interaction between development and UX.

**UX Review.** This theme reflects UX team involvement in the review of the software or ideas. Part of the challenge between UX and development was that the implementation was different from the design. The key difference in terms of user experience review is the extent to which UX is involved in validating the user experience of what is deployed. Whereas UX members in Team A play a significant role in making sure the product aligns with the design, this responsibility falls on the TPM in Team B.

**UX Inclusion.** This theme reflects the extent to which UX is included in the software development process. Both teams experienced the issue of developers not attending formal user experience meetings (e.g., retrospectives). However, it appeared that Team A was more inclusive of user experience than Team B. For example, participants noted being consulted on Team A, whereas opportunities for consultation ended on Team B (i.e., with the discontinuation of development demos).
**Developer Openness to UX.** This theme reflects comments or observations about how open development is to UX activities and professionals. Developers on Team A seemed to be more open to UX than developers on Team B. Although some level of openness is required by both to implement UX ideas, developers on Team A appeared to consider the UX perspective even if or when they weren’t required to do so.

6.2. Cross-Case Results

Thresholds play an important role in QCA interpretations. They determine what is considered high versus low levels of conditions. In this research, the within membership thresholds of conditions were set using the mean of survey responses to the measure (e.g., trust). The mean was used to offset bias in the responses. By setting the within membership threshold as the mean, what was considered the high presence of a condition better reflected what was above the norm (i.e., high relative to the norm of responses).

Density, as a measure of the structural dimension, was initially included in the QCA analysis as a causal condition. However, given it was a network measure and analyzed at the network level (instead of the individual level), it did not provide sufficient variance to contribute to the QCA analysis. It did, however, provide insight about the cohesion within the team as a point for consideration during overall interpretation of the context in which the teams collaborated.

**QCA Results**

Data from 15 UX and development participants, across cases, were used in the QCA analysis (3 UX designers, 12 developers). The average number of years developer or user experience participants spent on their primary project was 11.5 months (SD = 8 months). All UX and developer participants (except 1 missing response) agreed or strongly agreed that they were satisfied with their job (mean= 6.4, SD= <1). All UX and developer participants (except 1 missing response) somewhat agreed to strongly agreed that they were satisfied with their functional team (mean= 6.5, SD= <1). All UX and developer participants somewhat agreed to strongly agreed that they were satisfied with their project team (mean= 6.5, SD= <1).

QCA results are communicated and compared in terms of consistency and coverage. Consistency and coverage also facilitate the assessment of how well a formula fits the cases. Consistency answers the question of how often a given combination of conditions is associated with a specific outcome (i.e., frequency of the combination of conditions). Coverage answers the question of how many instances of a
given combination of conditions explain the outcome out of the total number of combinations that explain the same outcome (i.e., impact of the combination of conditions). Whereas raw coverage includes the overlap between causal paths in a solution, the unique coverage indicates the impact of a specific causal path as part of the overall solution. Unique coverage is equal to the solution coverage minus the raw coverage of all other formulas (or causal paths). After finding the unique coverage for each formula in the solution, the remaining difference between the solution coverage and the sum of unique coverage is equal to the overlap in coverage between formulas [106].

**Knowledge Construction**

During the knowledge construction phase, ideas can surface through brainstorming sessions, meetings, and other means of engagement among team members [32]. There were two causal recipes for user experience knowledge construction. Neither recipe provided insight about what conditions lead to the presence of knowledge construction but, instead, what would impede the presence of knowledge construction.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Interpretation</th>
<th>Consist.</th>
<th>Raw</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>~km.construction = ng.shared * sc.cognitive</td>
<td>The presence of a shared governance form and high shared meaning leads to the absence of high knowledge construction.</td>
<td>.87</td>
<td>.29</td>
<td>.07</td>
</tr>
<tr>
<td>~km.construction = sc.relational * sc.cognitive</td>
<td>The presence of high trust and high shared meaning leads to the absence of high knowledge construction.</td>
<td>.81</td>
<td>.56</td>
<td>.34</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td></td>
<td><strong>.83</strong></td>
<td><strong>.64</strong></td>
<td></td>
</tr>
</tbody>
</table>
two paths explain 64% of the cases with the absence of high knowledge construction (by developers and UX professionals).

Characteristics of shared governance are that governance is highly dispersed, and all (or most) members are responsible for the administrative aspects of the team and facilitate the team’s achievement of its goals [100]. Shared meaning (i.e., language) can make it easier for members to gain access to people and their information: enhancing the likelihood of knowledge exchange and combination [24,87]. Finally, people are more likely to cooperate with each other when trust is high [87].

These findings suggest that high shared meaning could make it difficult for UX and Dev to generate a high number of UX ideas given that it exists in both causal recipes for the absence of high knowledge construction. Given the lack of evidence about what conditions explain the presence of the perception of high knowledge construction, teams may want to diversify participation when constructing knowledge by including members/people that do not have the shared meaning, or they may want to engage in activities that create healthy conflict (e.g., competition).

Knowledge Embodiment

During the knowledge embodiment phase, ideas are captured in a container (e.g., a document) [32]. There were three causal recipes for user experience knowledge construction. None of the recipes provided insight about what conditions lead to the presence of knowledge embodiment but, instead, what would impede the presence of knowledge embodiment.

Table 8: Knowledge Embodiment QCA Solution

<table>
<thead>
<tr>
<th>Formula</th>
<th>Interpretation</th>
<th>Consist.</th>
<th>Raw</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>~km.embodiment = ng.lead * sc.cognitive</td>
<td>The presence of a lead governance form and the presence of high shared meaning leads to the absence of high embodiment.</td>
<td>.77</td>
<td>.42</td>
<td>.07</td>
</tr>
<tr>
<td>^km.embodiment = sc.relational * sc.cognitive</td>
<td>The presence of high trust and high shared meaning leads to the absence of high embodiment.</td>
<td>.88</td>
<td>.59</td>
<td>.12</td>
</tr>
</tbody>
</table>
In a lead governance environment with the perception of high shared meaning, team members may not embody knowledge if they expect the lead to do so or wait until the lead directs them to do so [86]. A lack of knowledge embodiment can also occur when there is a high level of trust and members are familiar with each other’s responsibilities. This may result from team members not considering embodiment as imperative since they trust each other: meaning knowledge remains tacit and, likely, moves to the latter phases. With shared governance, members may experience conflict and be disengaged from the network as a result of not having a perception of high trust, not being familiar with whose responsibility it is to embody the knowledge, and not taking action to ensure the knowledge gets embodied. The solution consistency of .73 means that 73% of the time these conditions are present, the outcome of high knowledge embodiment was perceived as absent by developers and UX professionals. The solution coverage of .84 means that these two paths explain 84% of the cases with the absence of high knowledge embodiment (by developers and UX professionals).

These findings suggest that the perception of high embodiment is unlikely when the perception of shared meaning is the same as the perception of high trust (i.e., when both are high or when both are low) if governance is shared. Given the lack of evidence about what conditions explain the presence of the perception of high knowledge embodiment, teams may want to use a lead governance form and foster high trust to achieve high knowledge embodiment when there exists shared meaning.

Knowledge Dissemination

During the knowledge dissemination phase, knowledge is made available to the team [32]. There were three causal recipes for user experience knowledge dissemination. None of the recipes provided insight about what conditions lead to the presence of knowledge dissemination but, instead, what would impede the presence of knowledge dissemination.
Table 9: Knowledge Dissemination QCA Solution

<table>
<thead>
<tr>
<th>Formula</th>
<th>Interpretation</th>
<th>Consist.</th>
<th>Raw</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sim\text{km.dissemination} = \text{ng.shared} \times \sim\text{sc.relational} \times \sim\text{sc.cognitive})</td>
<td>The presence of shared governance, the absence of high trust, and the absence of high shared meaning leads to the absence of high user experience knowledge dissemination.</td>
<td>.95</td>
<td>.19</td>
<td>.09</td>
</tr>
<tr>
<td>(\sim\text{km.dissemination} = \text{ng.lead} \times \sim\text{sc.relational} \times \text{sc.cognitive})</td>
<td>The presence of lead governance, the absence of high trust, and the presence of high shared meaning leads to the absence of high user experience knowledge dissemination.</td>
<td>.88</td>
<td>.19</td>
<td>.19</td>
</tr>
<tr>
<td>(\sim\text{km.dissemination} = \text{ng.shared} \times \text{sc.relational} \times \text{sc.cognitive})</td>
<td>The presence of shared governance, the presence of high trust, and the presence of high shared meaning leads to the absence of high user experience knowledge dissemination.</td>
<td>.95</td>
<td>.21</td>
<td>.10</td>
</tr>
</tbody>
</table>

| Solution                                     |                                                                                | .91       |     | .49    |

Similar to embodiment, there could exist conflict in the team as a result of not having high trust and not having high shared meaning about responsibilities. These challenges can lead to team members not disseminating knowledge or caring enough to ensure it gets disseminated. The perception of a lack of trust along with a shared understanding of responsibility can result in a team environment with conflict and disengagement by members that believe it’s another team member’s responsibility to disseminate knowledge. The lead is, then, likely consumed with managing the conflict and trying to achieve network goals, which can result in the lead not disseminating knowledge due to the increased effort in managing the team. These, collectively, can adversely affect knowledge dissemination. Finally, lack of knowledge dissemination can occur if the members trust each other, all (or the majority) of the members are engaged, and members share the meaning of each other’s responsibilities, so that each member has ensured they have the knowledge required for the team to achieve their goals. The solution consistency of .91 means that 91% of the time these conditions are present, the outcome of high knowledge
dissemination was perceived as absent by developers and UX professionals. The solution coverage of .49 means that these two paths explain 49% of the cases with the absence of high knowledge dissemination (by developers and UX professionals).

These results suggest that the absence of a perception of high knowledge dissemination only exists when there is both shared governance and the absence of a perception of high trust coupled with the absence of a high perception shared meaning. There is a lack of evidence about what conditions contribute to the presence of a perception of high knowledge dissemination. However, teams may be able to realize a perception of high knowledge dissemination by using a lead governance form and fostering a perception of high trust when there is the absence of a perception of high shared meaning, or by using lead governance and allowing the presence of a perception of high trust when there is the presence of a perception of high shared meaning.

**Knowledge Use**

During the knowledge use phase, knowledge provides commercial value (viz., ideas are incorporated into the product) [32]. There was one causal recipe for knowledge use, which did not provide insight about what conditions lead to the presence of knowledge use but, instead, what would impede the presence of knowledge use.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Interpretation</th>
<th>Consist.</th>
<th>Raw</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>~km.use = ng.lead * sc.cognitive</td>
<td>The presence of lead governance and the presence of high shared meaning of functional responsibilities in the team leads to the absence of high user experience knowledge use.</td>
<td>.80</td>
<td>.48</td>
<td>.48</td>
</tr>
</tbody>
</table>

An absence of a perception of high knowledge use may not occur when team members expect the lead to ensure ideas are incorporated into the build, or if team members are not aware that user experience knowledge has not been incorporated into a build. There is a lack of evidence about what conditions contribute to the presence of a perception of high knowledge use. However, the team may want to move
to a shared governance form if they are small, or use an administrative governance form if they are larger, to ensure that sufficient communication is occurring. The solution consistency of .80 means that 80% of the time these conditions are present, the outcome of high knowledge use was perceived as absent by developers and UX professionals. The solution coverage of .48 means that these two paths explain 48% of the cases with the absence of high knowledge use (by developers and UX professionals).

The following table (Table 9) provides a summary of the configurations that have the highest unique coverage. Since unique coverage reflects the proportion of cases that are uniquely covered by a given term (or configuration), this table indicates which configurations have the greatest impact on the absence of each outcome. For example, the first configuration (out of the two configurations related to knowledge management construction) has the highest impact on the absence of high knowledge management construction.

Table 11: QCA summary and impact table

<table>
<thead>
<tr>
<th>Formulas</th>
<th>~KM.C</th>
<th>~KM.E</th>
<th>~KM.D</th>
<th>~KM.U</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teams with high trust and high shared meaning</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teams with shared governance and high shared meaning</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teams with lead governance and high shared meaning</td>
<td>x</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teams with lead governance, absence of high trust, high shared meaning</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teams with shared governance, high trust, and high shared meaning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Solution Consistency | .83 | .84 | .91 | .80 |
| Solution Coverage    | .64 | .76 | .49 | .48 |

Note: Large bold O’s indicate which configurations had the largest unique coverage of all configurations in a solution.

6.3. Evaluation of Propositions

The study of agile user experience teams provided insight about the use of continuous integration and deployment as an alternative to iterative development. The primary difference between the two, based on observation during this study, is that iterative development aligns all deliverables to a schedule of iterations (or sprints) and releases (known as time-boxing). However, continuous integration and deployment removes this timing and deploys deliverables (e.g., features or fixes) as soon as they are done
and pass acceptance testing. The resulting effect of these differences was an inability to compare teams across iterations or releases as was originally planned. Hence, the findings shown below are discussed relative to the propositions to the extent possible. Gaps between the findings and the propositions are, then, provided as questions or propositions for future research.

**Description of the Current Situation**

It is important to understand the current interaction in agile user experience teams. Hence, the first proposition asserted that:

\[ P1: \text{It is anticipated, in a given project team and within a release, that all three governance forms (viz., the shared, lead, and network administrative) will be found, that social capital levels will vary at various stages of development, and that the level of user experience knowledge management will vary based on the governance form, the level of social capital, and the stage of development.} \]

The findings suggest that two of the governance forms—shared and lead—are used in an agile user experience team. The findings also suggest that the absence of high trust and high shared meaning are only related to the absence of user experience knowledge embodiment and dissemination. Finally, variation among governance forms and shared meaning appeared to affect how knowledge is managed. The combination of lead governance and high shared meaning are associated with the absence of user experience knowledge embodiment and use. The combination of shared governance and high shared meaning are related to the absence of knowledge construction.

Findings evaluated against the first proposition leave open questions about whether trust, cohesion, or network administrative conditions affect how knowledge is managed and what changes exist as a result of the stage of development. There was no evidence of whether user experience knowledge management varies based on the governance form, social capital, and stage of development since it was not possible to compare the teams based on development stages.

**Conditions within a Release**

The intention of the second proposition was to identify what combinations of conditions were used at different stages of development within a release cycle. This differed from the previous proposition in that it supposed that all combinations of conditions are used in a development release.
P2: It is anticipated that all combinations of social capital, governance form, and user experience knowledge management are employed at different stages of development within a release cycle.

Findings suggest that most conditions apply within each phase of knowledge management. All solutions—within each phase of knowledge management—entailed a form of governance and a condition for the cognitive dimension of social capital. Most solutions also included a condition for the relational dimension of social capital.

However, the relational dimension was not present in the knowledge use phase. Additionally, the solutions did not have results in terms of when an administrative form of governance is present or absent. The structural dimension was not informative given it was a network measure and was similar for the two participant teams. Hence, the structural dimension results were removed from the analysis given their homogeneity.

This leaves open questions about: what are comparable temporal units of analysis between teams that use an iterative development approach and continuous integration approach? What are the effects of time on the combinations of social capital, governance form, and user experience knowledge management? It was not possible to compare the teams by release since one team used a continuous integration approach to development and the other used an iterative approach. Hence, it was not possible to determine which combinations of conditions exist in various stages of development within a release.

Optimal Relationships among Theoretical Components
The intention of the following proposition was to identify which combinations of conditions were optimal toward achieving the desired user experience knowledge management in a given phase of a release cycle.

P3: It is anticipated that an optimal configuration for achieving the desired user experience knowledge management for teams in the beginning of a release is medium social capital and a shared governance form given the need to determine and agree on scope. It is anticipated that an optimal configuration for achieving the desired user experience knowledge management for teams in the middle of a release is low social capital and lead team governance given the need for a focus on implementation. It is anticipated that an optimal configuration for achieving the desired
Findings suggest which combinations could be considered optimal in terms of which combinations could impede high knowledge construction, embodiment, dissemination, and use. Specifically, these are: 1) teams with high trust and high shared meaning are most likely to experience an absence of high knowledge construction and embodiment; 2) teams with lead governance, an absence of high trust, and presence of high shared meaning are most likely to experience an absence of high knowledge dissemination; and 3) teams with lead governance and high shared meaning are most likely to experience an absence of high knowledge use. These combinations had the highest unique coverage in the solution for each phase of knowledge management, indicating that they have the most impact.

However, there was no evidence to suggest which configurations were optimal toward the presence of high knowledge construction, embodiment, dissemination, and use. Furthermore, there was no evidence to suggest which combinations of conditions are optimal at the beginning, middle, and end of a release since iteration and release data could only be collected from one team (thus, constraining the ability to make a direct comparison between the teams).

6.4. Framework Expert Review

Four domain experts were asked to provide feedback on the perceived benefits, limitations, application, and challenges with applying the framework, based on their areas of expertise. They were provided with materials related to the framework, and they were asked to respond to a series of questions about the framework. The evaluation covered four areas of inquiry: the perceived benefits, application, limitations, and application challenges of the framework. Follow-up questions were used to resolve discrepancies.

Participants reported such benefits as the framework’s potential to better integrate team members of different backgrounds and improve the productivity and performance of the team.

In terms of application of the framework, the experts reported that using scenarios as guidance would be useful in applying the framework, and that the framework could be applied in teams where there is uncertainty in the team or innovative leadership willing to take risks. It was also communicated that
techniques such as workshops, tutorials, consulting, and discussions were effective methods for applying the framework.

Participants reported the following limitations of the framework: the possibility of the team experiencing gridlock if they were highly polarized, that the framework should be more accessible to practitioners given its currently theoretical nature, that it should also appeal to agile developers, that it should provide actionable guidance for the practitioner, that there might exist conflict between the goals of the individual team members and the team, and that the framework should be sensitive enough to recognize influences that are not explicitly part of the model.

The limitations noted are consistent with the vision for the framework. The current implementation of the framework is intended to help us understand existing interactions in a multidisciplinary agile user experience team. The insight gained from studying teams with the existing model will help us identify additional factors that might influence how user experience knowledge is managed and inform the crafting of practical guidance on how team members can improve their interaction and minimize polarization. Additionally, user experience knowledge management is the outcome for the current model, but it is our goal to gradually expand the area of concern to other roles toward improving knowledge management throughout the team generally.

Key challenges to applying the framework mentioned by the reviewers were obtaining buy-in by management and team members, and motivating practitioners to utilize the framework over existing approaches. Also mentioned were challenges with ensuring effective communication and coordination in the team, and balancing short-term and long-term strategic interests.

Adoption is a common concern when introducing tools or approaches, and we acknowledge these potential challenges. Our aim is to increase the likelihood of adoption by building the framework on an empirical foundation. By studying teams using the existing model, we anticipate the model will ultimately better reflect the actual concerns of multidisciplinary agile user experience teams. Informing the framework with empirical data from agile user experience teams will also help establish buy-in from practitioners. We also anticipate that keeping agile principles in mind during the development of the framework will help toward mitigating application challenges.
6.5. Chapter Summary

This chapter provided a summary of results for each case, findings from the framework evaluation via QCA, discussion based on the evaluation of the results against the propositions, and results from the expert review. Discussion of the QCA results was combined with the results as a means to keep the causal recipes tightly associated with the interpretation and case details for ease of understanding.
Chapter 7: Team Interaction Framework (Revisited)

Chapter Outline:

Team Interaction Framework
Implications for Practitioners
Agile User Experience Team Social Interaction Practices
Chapter Summary

After reading this chapter, you should:

- Understand what the findings mean in light of the Team Interaction Framework
- Understand what the findings mean for the various actors of the team
This chapter contains generalizations to the theory based on the findings. This section also contains implications for practitioners of agile user experience teams.

7.1. Team Interaction Framework

This chapter relates the findings back to the Team Interaction Framework. Specifically, it considers the relevance of each component in light of the results.

Social Capital

Social capital was measured via the structural dimension (i.e., density), relational dimension (i.e., trust), and the cognitive dimension (i.e., shared meaning). Social capital provided key understanding about the amount of cohesion among team members, the trust between developers and designers, and the level of shared meaning between designers and developers. It is unclear to what extent the structural dimension is necessary in evaluating the framework (via QCA) since it appears to be within a close range for agile teams generally (which could provide insight as a control variable). Hence, the most relevant social capital information (with the most diversity) appears to be the level of trust among developers and designers, and the extent to which team developers and designers have shared meaning.

Network Governance

There appeared to be some differences of opinion among developers and designers about whether the team used a lead or shared governance form. Such variability about how decisions are made suggests that the governance form used is not a given as one might expect in an agile setting since agile promotes a democratic structure. Network governance also appeared to play a more significant role in knowledge dissemination and use. In essence, governance form was part of a configuration that explained the largest amount of coverage for knowledge dissemination and use. The presence of governance as part of those configurations suggests that network governance has import (but less import for knowledge construction and embodiment).

User Experience Knowledge Management

Understanding how user experience knowledge is managed helps to link team dynamics to the idea of users’ experiences of the product without having access to users. In other words, one could suggest that if input from UX professionals is incorporated, then users are likely to have a good experience. Also, there appeared to be diversity in the configuration of social capital and network governance conditions related to the absence of each phase of knowledge management. Hence, measuring how user experience knowledge is managed is a key outcome for the theoretical framework.
7.2. Implications for Practitioners

The findings from this research have implications for how practitioners influence the social interaction between developers and UX professionals while performing their tasks (and as a team in general). The following sections explicate these implications.

User Experience Professionals

User experience professionals have the challenge of advocating for the inclusion of user experience ideas (with varying levels of success). Findings from this research suggest that UX professionals are more likely to have UX ideas implemented when they have an intermediary that, at least, sympathizes with UX and has the respect of developers. This respect is likely to stem from the intermediary having past experience with, or simply intimate knowledge of, development principles and activities.

The findings also suggest that UX ideas are more likely to be implemented when UX professionals include development throughout the design process and enforce design ideas (e.g., wireframes and mockups) after implementation. Including development means, for example, providing development an opportunity to give feedback on a design before it is completed. Enforcing design ideas would mean holding development accountable by providing a pass/fail response once the design has been implemented depending on the extent to which the feature honors the design.

Software Developers

Software developers are the ultimate arbiters of what gets implemented into the product. However, they are still part of a team that is collectively responsible for the success of a product. The findings suggest that developers may not necessarily be inclined to include UX in their activities or heavily participate in UX activities. This can create a barrier between development and UX since it would take more time to come to a shared understanding of how members in each role think.

Architects and Managers

Technical and staff managers have a responsibility to coordinate and control resources toward achieving product or project objectives. The findings suggest that managers play a significant role in the extent to which UX and development interact. For example, the architects and product managers can determine to what extent the system will account for UX ideas, whereas developer leads and project managers can directly influence to what extent developers and UX professionals will interact. In this research, the role
of the Technical Product Manager (TPM) appeared as a key determinant of how much development and UX directly interact.

**Agile User Experience Software Teams**

The interaction between UX professionals and developers is subject to influence by members in other roles on the team. Consider, for example, the findings about UX supplementing the Quality Assurance (QA) role by enforcing design implementation. This inclusion of UX resulted from an unclear understanding within QA about whether to test against the design or the implementation. Hence, roles such as QA and documentation can, intentionally or unintentionally, either facilitate or hinder the interaction between UX professionals and developers.

### 7.3. Social Interaction Practices for Agile User Experience Teams

Based on the findings of this research, several social interaction practices are recommended for agile user experience teams. Although these practices are specific to developer and UX relations, they have import for the entire team as well.

**Enable healthy conflict.** Healthy conflict can help teams filter ideas and promote creativity by encouraging members to think beyond the surface. When team members are generating user experience ideas and capturing them in some tangible form (e.g., a wireframe), they are engaging in a creative process. Hence, team members challenging initial ideas and designs can help them think more deeply about the users and their possible experiences. Teams can achieve healthy conflict by including members that lower the average trust and shared meaning in the team. For example, when holding a review meeting, teams may attempt to exclude members they don’t trust or don’t think understand their roles. However, these members should also attend to generate healthy conflict. Ground rules can be established and enforced to ensure the conflict remains healthy. One instance of facilitating healthy conflict in the case was allowing user experience designers to enforce alignment of the implementation to the design.

**Strategically employ governance forms.** Shared decision-making can help members feel included and part of the team. However, lead decision-making can curtail coordination overhead when only one or a few members can make a decision. For example, during the dissemination of user experience knowledge, team members are making knowledge accessible to other members. Lead decision-making could achieve the goal when there is high trust and familiarity with each other’s roles. Hence, having one or a few members lead can ease the coordination of knowledge dissemination. Alternatively, the team can
designate an external entity (e.g., manager or project management office) to bear the responsibility for disseminating knowledge. The use of the Technical Product Manager (TPM) in the case study could be considered an example of how network administrators could reduce the overhead if their role was more broadly defined.

**Encourage shared ownership of the product.** The final product deployed to the user is a result of the effort invested by members belonging to a variety of roles on the team. However, developers have the most direct influence since they write the code. By encouraging shared ownership of the product, team members are held accountable for ensuring that each role’s contribution is duly considered. Teams can encourage shared ownership during implementation by fostering discussions between user experience and development during the implementation of design ideas. One example from the case is the developer getting input from the user experience designer during implementation. At this moment, the designer was able to share her knowledge about visual perception with the developer to help with translating the design into implementation more effectively.

### 7.4. Chapter Summary

This chapter considered how well each component fits within the Team Interaction Framework after conducting the case studies. Each component appears to fit within the framework, but there’s room to study which measures are most appropriate for each component as future work. This chapter also provided three key practices that teams can employ as a means to improving their social interaction.
Chapter 8: Framework Application Guide

Chapter Outline:

Steps for Conducting the Study

Study Components

Chapter Summary

After reading this chapter, you should:

● As a practitioner, know how to apply the Team Interaction Framework to achieve the desired goals within your agile user experience team
This chapter shows a practical representation and description of the framework along with a scenario illustrating how teams can use it. This chapter also provides recommendations on how agile user experience software teams can apply the Team Interaction Framework to influence their interaction based on the findings.

This chapter provides the basic steps required to perform an analysis using the Team Interaction Framework as a quick guide. The steps are provided, and are followed with details about what each step entails in the subsequent sections.

8.1. Steps for Conducting the Study

1. Decide on a Network Boundary
2. Prepare for the Study
3. Collect the Data
4. Analyze your Results
5. Report your Findings

You’ll first want to define your network boundary. This first step is important in defining the scope of the study and its participants. Identifying which members are part of the team (or network) has implications for how you identify participants and the findings. For example, including a member that is not part of the team could skew the density and centralization results. One way to establish the network boundary is to request of list of project or product team members.

8.2. Study Components

Study Preparation

Preparation for conducting research is just as important as its execution. Gaining access and approval to conduct the study, ensuring the materials are ready for use, determining which data collection and analysis methods you’ll use, and deciding how to deliver the findings are important to consider relative to the study environment prior to execution. The following sections provide guidance on how to prepare so you can get the most out of a team interaction study using the Team Interaction Framework.

Organizational and Team buy-in

Gaining access or approval to study teams can be arduous. Executives and management can perceive the study as detracting from employee focus (and profitability) or exposing negative characteristics about
their organization. Employees may see the study as jeopardizing their security in the company or as an interruption to their flow. These concerns can challenge the effectiveness of the study and have implications for how the research is designed and what level of preparation is required to conduct the study.

The Team Interaction Framework is already designed to require a minimal amount of effort and intrusion, but several key strategies are provided here as reminders to help with gaining organizational and team buy-in: limit the time required to conduct the study, seek a referral or advocate to help gain access and obtain data, and/or ensure the contribution to the organization or team and risks are clearly communicated.

It is often easy to accept that gaining access is not in the researcher’s control. However, by limiting the time required to study the organization, the researcher is helping to alleviate concerns about productivity and focus. As an example, consider hearing that a study required 2 hours of your time versus 15-20 minutes of your time. When potential participants feel they can contribute in a reasonable amount of time, they are more likely to make time to participate as they meet their job requirements. For executives and management, it is helpful to communicate the time required as the sum of all activities (e.g., 20 hours) as opposed to the amount of time required of each participant.

Seeking a referral or advocate can also help with gaining access and obtaining data because people are sometimes more accepting of requests from a colleague. Also, an advocate in the organization or team is likely able to communicate the interests in a language consistent with the organizational or team culture. It is helpful to provide the advocate with material, such as a flyer, they can distribute or talk to when communicating the research. Having an advocate is one of the more effective ways of gaining access, but may also be among the toughest to employ since the advocate’s reputation is at risk.

Finally, it is important to ensure the organization or team knows what they will gain from the study and what amount of risk is involved. The goal is to help the organization or team to estimate the impact of the study. By communicating explicitly, the researcher is able to either obtain more support for their study, or realize early that a given team or organization may not be a suitable environment.
Materials
The Team Interaction Framework contains the minimal materials required to assess the team’s social interaction relative to how they manage user experience knowledge. Specifically, the framework entails the use of a questionnaire to collect data about: the behavioral dynamics of members on the team; the structural (viz., the collaboration, help, and advice relations), relational (viz., trust), and cognitive (viz., shared meaning) dynamics; and the degree to which members on the team can influence the team’s direction.

As part of preparation, the researcher may want to consider whether the instruments measure what they are interested in understanding. For example, knowledge management questions are currently tailored to user experience knowledge. However, these can be modified to account for other forms of knowledge on the team. Also, to measure the structural dimension, the tables should be populated with members of the team being studied. These names will also define the network boundary [51].

Finally, the researcher may want to add materials to provide richer data or capture more data. For example, the Team Interaction Framework primarily relies on a questionnaire and notes from observations and interviews. However, there are no fixed questions for artifacts to guide the researcher. If the researcher will lead a multi-researcher effort, they may need to provide more structure in the form of specific interview questions or guidelines, observation guidelines, or artifact review guidelines. The researcher should keep in mind, however, that each artifact could contribute to the study requiring more effort from or intrusion to the participants. For example, asking participants to identify and interpret artifacts can burden a participant cognitively and in terms of their productivity.

Research Design
There are various ways to design research, depending on such factors as the research question being answered, access to data, and the kind of data available. The Team Interaction Framework employs a case study design [125] where each team is considered a case. Each case entails background description, analysis results and interpretation from the components of the framework, as well as implications and recommendations for the team. The remaining sections of “Conducting the Study” are in the context of a case study research design.
Research Question and Propositions

Research questions (and sometimes propositions or hypotheses) are conventional starting points for any research study. Case study research questions ask “how” or “why” about some social phenomenon [125]. The overarching research question for the Team Interaction Framework asks how the social capital and network governance form influence the way user experience knowledge is managed in agile user experience software teams. Which participants are studied will determine the scope of interaction (e.g., between developers and user experience professionals).

Participants

The Team Interaction Framework primarily focuses on the interaction between developers and user experience professionals. However, with the aforementioned modifications to the materials, the framework can accommodate other participants on software teams.

In terms of the social network analysis, sample and participant decisions influence what is ultimately considered the network boundary. This network boundary distinguishes who is considered a member of the network and who is not. In the Team Interaction Framework, the network boundary is set for the project: those that are members of the project or product team are considered members of the network.

Validity

Validity is concerned with ensuring that the researcher is actually measuring what they intend to measure. For example, asking a participant whether they like collaborating with another team member does not necessarily measure whether they like the team member. However, the researcher can take steps to ensure the findings tell more of the story to corroborate the link between liking to collaborate with a team member and liking the team member. Yin presents three forms of validity in case studies and strategies for strengthening validity [125]. Only simple, but effective, tactics are presented here.

Constructs

Constructs are concepts that are being studied. Construct validity, then, is concerned with ensuring the measurements taken by the researcher actually measure the constructs (or concepts). In the Team Interaction Framework, the concepts are social capital, network governance, and knowledge management. One of the data collection tactics for increasing construct validity suggested by Yin is to use multiple sources of evidence. The Team Interaction Framework uses interviews, observations, and surveys as its sources of evidence (also known as triangulation). One benefit to triangulation is the decrease in the
subjectivity of findings if results converge from the three sources. Hence, it is important to maintain at least three sources of evidence.

The second type of validity is internal validity (or causal). Concluding that one thing causes another (i.e., x causes y) requires showing that nothing else could have contributed to x causing y. One simple and effective way to identify whether the causal relationship is the result of some other factor, suggested by Yin, is to address rival explanations (and confirm that the causal explanation is supported through multiple sources of evidence) by thinking through what other reasons some outcome could have occurred given the evidence. The Team Interaction Framework evaluates causal relationships via the Qualitative Comparative Analysis, but this same thought process is required when performing this analysis (to identify what Ragin calls counterfactuals [108]).

The third type of validity is external validity, which is concerned with determining whether the findings are generalizable beyond the current case. It may be difficult to convince others of the relevance of findings to other teams based on a single case (or instance) of how another team interacts. When there is a single case, Yin suggests using theory and generalizing to that theory. Multiple cases provide support for generalizations much like running multiple experiments when testing a hypothesis. The Team Interaction Framework addresses external validity since it is grounded in theory. The researcher can provide further convincing evidence by studying multiple teams or a team multiple times.

**Reliability**

When conducting a study, it is important that other researchers can conduct the study the same way it was conducted when previous findings were reported. In doing so, other researchers should get similar results. Yin suggests using a case study protocol that researchers can follow later, and creating a database of the case studies. The Team Interaction Framework provides a case study protocol, which would need updating if modifications were made to the study. The framework also accommodates creating a case study database since each team is a case. In essence, the researcher can organize, filter, and sort the case studies in the database by fixed characteristics of the framework (e.g., cases with high social capital, cases with shared network governance) to help them identify patterns across cases.
Data Collection Methods

As mentioned previously, the Team Interaction Framework utilizes three sources of evidence toward understand the impact of social interaction on knowledge management in agile user experience teams. Specifically, these are the survey, observation, and interview.

Questionnaire
The purpose of the Team Interaction Framework questionnaire is to collect data from all team members on the theoretical components. It essentially provides an opportunity for the respondents to answer questions on their own time in the absence of the researcher. The sections of the questionnaire measure: the state of the project (e.g., what iteration it is in); participant job satisfaction; the structural, cognitive, and relational dimensions of social capital; the form of network governance; and how user experience ideas flow through the knowledge management phases.

Interview
The purpose of the interview is to obtain richer data about the theoretical components. Interview questions can stem from questionnaire responses, observations, or knowledge about the theoretical components. Interviews can require a significant amount of participant time and effort, making them difficult to sell as a need, to participants and their leadership. Depending on the time constraints, it is possible to interview everyone on the team or to perform spot interviews of key participants as needed. The most important consideration is that sufficient interview data is collected to serve as corroboration of observations and questionnaire responses for the given research question.

Observation
Observation is one of the more credible research techniques because evidence is seen directly by the researcher. Observations also provide an opportunity for researcher grounding when analyzing and interpreting the findings. Although the Team Interaction Framework does not provide artifacts to guide observation, the researcher can develop artifacts to aid in their recording of key moments based on the theoretical framework. It’s important to note that observations are still subject to researcher bias and the multiple sources of evidence help to guard against overly biased findings or threats to study validity (e.g., interviews can help the research put what they see in context).

Data Analysis and Interpretation
Quality data collection can be compromised if data analysis is not performed rigorously. This means ensuring that the assumptions and limitations of the methods are understood, and following the steps
with prudence. The Team Interaction Framework uses established analytical techniques that can be learned in more detail from original sources. Hence, this section will only provide enough detail necessary to perform the analyses.

*Social Network and Statistical Analyses*

The social network analysis provides insight about such team characteristics as the level of trust among members, how cohesive the team is, and to what extent they have shared meaning. Following are the methods for calculating and interpreting social capital within a team.

**Structural**

The structural dimension of social capital measures the density (or cohesion) and degree centralization (how much the team is dependent on one or a few members) within a team. The structural dimension data is collected via questionnaire for the collaboration, help, and advice relations (defined on the questionnaire). Responses from all team members should be included in the calculation. Although it’s possible to calculate the density and centralization by hand using common formulas \[123\], it is recommended that you use social network analysis tools (e.g., R \[119\], UCINET \[13\], ORA \[19\], Gephi \[8\]). The formulas can provide understanding about the calculation and the meaning of the result, but the tools can save the researcher a considerable amount of time and become standardized through use by a large number of users.

**Cognitive**

The cognitive dimension of social capital is measured, in the Team Interaction Framework, as the extent to which members perceive they are familiar with each other’s functional responsibilities on the project team in the form of a 5-point likert scale of agreement. These responses are averaged across participants of interest (e.g., developers and designers if their interaction is under study) to determine the cognitive dimension score. This score is then directly interpreted relative to the agreement scale. Most important in the interpretation is noting the average amount of shared meaning among participants. However, individual scores can be used to identify anomalies that might help inform discussion of findings from multiple sources.

**Relational**

The relational dimension of social capital is measured, in the Team Interaction Framework, as the level of trust members have generally, in their functional team (i.e., domain), and in their project/product team members. Trust is measured through a 5-point likert scale of agreement. These responses are averaged
across participants of interest (e.g., developers and designers if their interaction is under study) to
determine the relational dimension score. This score is then directly interpreted relative to the agreement
score. It is important to note that the question about general trust is reversed. This means that the points
should be inverted when calculating the relational scale. Most important in the interpretation is noting
the average amount of trust among participants. However, individual scores can be used to identify
anomalies that might help inform discussion of findings from the multiple sources.

The relational dimension is also measured via the agreement on norms and values among team members.
This is measured in the Team Interaction Framework by having team members rank the top three norms
and top three values they perceive within the team. The data is analyzed using Krippendorff’s alpha [90]
or some other inter-coder agreement method (e.g., Cohen’s kappa) where characteristics of the data align
with the assumptions of the statistical technique. Interpretation of the results is dependent on which
method is used.

Network Governance

Network governance helps researchers understand how network (or team) decisions are made (i.e., which
members have a say in the administration of the network). Network governance is measured in the Team
Interaction Framework by finding the level of agreement among participants about whether the team
uses a shared, lead, or network administrative form. Agreement is measured using the following formula:

\[
\text{Agreement} = \frac{\text{# of observed pairs of agreement}}{\frac{n(n-1)}{2}}
\]

\[ n = \text{# of raters} \]

(Note: the number of observed pairs can be calculated using \(n(n-1)\)
where \(n = \text{the number of the responses or ratings being analyzed}\)

The interpretation of the result is the percentage agreement within the team. This formula accounts for
multiple agreements versus just one selection (e.g., the number of members that agree on shared plus
the number that agree on lead, as opposed to just calculating a simple majority agreement). Hence, 30%
agreement could mean some participants agreed that a shared governance form was used and some
agreed that a lead form was used.)
Knowledge Management

The knowledge management score helps researchers determine the extent to which participants perceive knowledge being constructed, embodied, disseminated, and used within the team. Knowledge management is measured in the Team Interaction Framework in the form of a likert response scale. The results are analyzed by taking the mean of the likert response across the participants under study for each phase of knowledge management.

Thematic Analysis

A thematic analysis is a basic qualitative analysis approach where themes or concepts are identified from rich data collected through such means as interviews and observations. The themes are then synthesized or condensed into more significant themes until there are approximately 3-5 themes. The themes are then interpreted relative to the research question and incorporated into the discussion. The thematic analysis can provide a rich perspective, which is a challenge with the other analytical approaches.

Qualitative Comparative Analysis

Qualitative Comparative Analysis (QCA) enables the researcher to make causal inferences about the theoretical framework. The conditions (or variables) are entered into the data table as fuzzy values and the result is a formula (or model) about which conditions are likely to lead to the outcome of a given knowledge management phase being studied. For example, each row (or case) could have the following as values in each column: network density, average trust score, average shared meaning score, network governance form, and the outcome of interest (e.g., average knowledge management construction score). Interpretation is communicated in terms of necessary and sufficient (taking into consideration the consistency and coverage of each model and the overall solution), and should occur in the context of case details or theory.

Reporting

When reporting the findings, it’s important to provide enough context for the team to understand the findings. This can be accomplished by including the following sections:

1. **Executive Summary.** Providing a summary of the problem, approach, and findings can provide insight into the team’s dynamics quickly. This is especially important not only for executives, but also team members that might not read through the report, but want a rough idea of how their team fared.

2. **Introduction.** The introduction should include the motivation for the research, its purpose, questions or propositions it will address, and the significance of the study.
3. **Background.** The background section should provide a description of the company, project team, agile or lean environment and practices, development team practices, UX practices, and the interaction between development and UX.

4. **Methodology.** Providing insights about the methodology helps inform team members how their social interaction was assessed and how to better make sense of the findings. It should provide information about what forms of data collection and analyses were used, key methodological decisions such as how significance was determined and how thresholds were set, and relevant rationale for those decisions.

5. **Results.** The results section provides the concise results from the social network analysis, thematic analysis, and any statistical techniques that were used. This section should provide the results along with their interpretations.

6. **Discussion.** The discussion section is where the results (and their interpretation) are then communicated relative to how each role on the team does their job. It, then, may be useful to provide any recommendations for what team members in those roles could do toward improving the social interaction on the team. It might also help to provide limitations to the study as a way to further help team members (and other readers) evaluate the findings.

7. **Conclusion and References.** Finally, the report should provide conclusions that summarize the findings and references to any sources used as part of the study.

8.3. **Chapter Summary**

Typically, practitioners do not have (or do not want to invest) the time to develop a framework and related materials when they want to conduct a study. This chapter provides a quick guide to studying a team’s interaction using the Team Interaction Framework and its materials. This chapter provides a perspective on the various components considered important in research to explicate the assumptions the practitioner would be adopting when using the framework.
Chapter 9: Conclusions and Agenda

Chapter Outline:

- Contributions
- Limitations and Future Work

After reading this chapter, you should:

- Understand the contributions of this research in HCI
- Understand the limitations and future agenda of this work
The preceding chapters argued that there is a need for social integration within agile user experience software teams because of the impact social interaction has on the team’s output. Integrated processes, the sharing of practices, collaborative technology, and team integration have been considered as useful approaches. However, the integration that is implicit in each of these areas, but least addressed, is social integration.

9.1. Contributions

The contribution of this work is three-fold: (1) an empirically substantiated Team Interaction Framework for understanding and changing the social interaction in agile user experience teams, (2) an instrument and method for assessing a team’s social interaction and user experience knowledge management, and (3) practices for improving the social interaction in agile user experience teams based on the framework. The instrument and method used in this research were developed to measure social capital, network governance, and how user experience knowledge is managed with the intention of the instrument and method growing in use as a continuous means to assess the social interaction in agile user experience teams. Case studies provide guidance to practitioners on how to use the instrument and method. The Team Interaction Framework provides software teams with a means to interpret the dynamics on their team that can be used to diagnose interaction challenges and strategically address them. The Team Interaction Framework also serves as the foundation for strategies to systematically improving the social integration in their teams and, thus, requires less overhead in identifying and addressing challenges efficiently and effectively. The social interaction practices are actionable, minimally risky to apply, and are lightweight in terms of the effort required to apply them.

9.2. Limitations and Future Work

A diversity of integration approaches has contributed to the integration of the software development and user experience communities on agile software projects. However, research on social integration fills a need to improve the social capital among software developers and user experience professionals that is superficially and indirectly considered in some of the other approaches. This research sought to contribute toward satisfying this need through studying how social capital and network governance relate to the management of user experience knowledge. With this understanding, insight is gained about how social interaction can influence a software team, which has implications for the software product.
Limitations and Constraints of the Study

Subset of actors, activities, outputs, and approaches
As a means to focusing this research, a subset of actors, activities, outputs, and approaches were considered. For example, UX professionals and developers were the primary actors of interest. User experience was the primary domain of interest, and knowledge management was the primary activity of interest. Outputs, such as wireframes and mockups, were not reviewed given the constraints related to using proprietary data. Only agile and lean development project environments were part of this study. Finally, this research was concerned with the social interaction in agile user experience teams, but the social interaction in the context of the other integration approaches is also important to explore. There are many other actors, activities, outputs, and approaches that are suitable for future work.

Preliminary framework evidence
The use of Qualitative Comparative Analysis was strategic given the anticipation of having a low number of participant teams. This use of QCA yielded results about what conditions would lead to the absence of the outcome. However, more cases would be required to identify what conditions might explain the presence of the outcome. Although not necessarily a limitation in the context of scientific findings, practitioner teams might value also knowing what conditions lead to the presence of user experience knowledge management outcomes.

Relatedly, an inherent challenge of research is getting full participation from the desired participants. A challenge in this study was that a subset of the full team participated in the study to varying degrees. Although obtaining full participation from a team can help with better understanding how each team member perceives the interaction in the team, it is possible to gain sufficient understanding from a diverse subset of team members.

The Team Interaction Framework was developed with practitioners in mind, but is currently more theoretical than practical given the nature of the research. Although it is common for scientific research to start theoretically, the next step is to make the framework more consumable by practitioners through refinement based on lessons learned from ongoing application of the framework in team settings.

Theoretical robustness
There are a variety of ways to measure phenomena related to the theoretical components of the Team Interaction Framework. However, only measures required to achieve the goals of this research were used.
For example, trust was used as the primary measure for the relational dimension of social capital. Combining trust and norms into a complex indicator might provide additional insight. Hence, there is room to explore whether other measures (or their combination) are more suitable for studying the social interaction in teams when using the Team Interaction Framework to make the framework more robust. Additionally, each theory has its inherent limitations that also apply to their use in the Team Interaction Framework. Theories will improve and additional measures will become available as understanding about social interactions evolves in the respective research communities.

**Future Work Agenda**

The contributions of this research serve as a starting point for stronger systematic consideration of the social interaction in agile user experience software teams. The emphasis in this research was on making the argument that social interaction matters because it can impact what the team produces. Further research is needed to provide teams with a standardized means of assessing their social interaction and making strategic adjustments. Toward achieving these objectives, future research would aim to:

**Teams**

1. **Study the effect of social interaction in agile user experience teams relative to other actors, team activities, and outputs.**
   a. This study focused on understanding the social interaction between user experience professionals and software developers. There is value in considering the social interaction among those in other roles on the teams (e.g., project/product managers, documentation staff, and management) and those external to the team (viz., users and customers). The interaction between such roles should be studied in isolation as well as in conjunction with each other to understand the impact social interaction changes could have across relationships. This additional data would also enable the comparison of QCA results with other analytical approaches such as statistical analyses. For example, how does the social capital between user experience professionals and project/product managers influence the outcome of interest for agile user experience teams?

   b. Understanding how user experience knowledge is affected by a team’s social interaction provides one perspective. Other perspectives worth investigating include how social interaction influences the user’s actual experience and project factors (viz., cost, quality,
and scope). For example, might better social interaction within the agile user experience software team mean a system with fewer bugs?

2. **Study the role of social interaction in other integration and project environments**

   a. Agile user experience teams distinguish their approaches from traditional approaches by removing some of the traditional practices (e.g., big upfront design) that impede a team’s ability to remain consistent with the principles of agility. However, some traditional practices might fit well in agile user experience teams depending on the team’s social interaction and environment. For example, might a government client require an agile user experience process that can accommodate a big upfront design? How can improving the social interaction make this traditional practice more agile?

   b. The primary goal of agile user experience software teams is to produce quality software that is delivered on time, within budget, and provides a good user experience. Social integration is but one approach to accomplish this goal. However, a hybrid of social integration and other forms may provide a greater benefit, overall. Answers could be sought to a number of questions. For example, which combination of the integration approaches yields the greatest benefit for agile user experience software teams? In what situations do the various combinations contribute the most or least?

   c. It can help with developing strategies for improving the social interaction in agile user experience teams by enabling teams to investigate how to tweak specific practices to achieve better social interaction. For example, teams might inquire about what measurable effect does pair programming or daily stand-up meetings have on a team’s social interaction? Instead of these being standard practices throughout the entire lifecycle, should they only occur in certain points of the lifecycle? If so, in what moments do they most contribute (or detract) from the team’s social interaction?

Framework

3. **Help teams better integrate the framework into a larger strategic context and become more practitioner friendly.** Practitioners use a variety of methods to plan and monitor their efforts. For example, teams and businesses often use a SWOT analysis [99] to identify strengths, weaknesses, opportunities, and threat as part of strategic planning. A balanced scorecard system is used to
capture key metrics the teams or businesses will monitor as they execute their plan. The goal is to improve the Team Interaction Framework such that practitioners can incorporate insight about the social interaction in their teams in their SWOT analysis and scorecards. By doing so, the Team Interaction Framework becomes an integral part of the team’s process of assessing how their social interaction influences the value they seek to deliver to their customers and users.

**Theory**

4. **Study the use of various measures in the Team Interaction Framework.** This research defined the major components of the Team Interaction Framework. However, there are subcomponents and alternative ways of collecting data that could provide additional insight about the social interaction in agile user experience teams. For example, trust is one measure of the relational dimension. However, norms and values are as well. The questions, then, are: whether trust, norms, and values are required measures; is trust the most important measure; or is there a different measure that is more appropriate? As more evidence surfaces, better social interaction practices can be provided to teams. As theories improve, they can be applied to the Team Interaction Framework. The longer term goal is to scale the Team Interaction Framework to a broad number of agile user experience teams such that it becomes a standard practice in agile user experience software teams.

It is no secret that the stereotypical programmer tucked away in a closet without social skills is out of style. Processes, technology, and practices are important, but social dynamics need focus as well. This research attempted to shine light on the social interaction of developers and user experience professionals on agile user experience teams.
References


57. Hussain, Z., Slany, W., and Holzinger, A. Investigating Agile User-Centered Design in Practice: A Grounded Theory Perspective. *5th Symposium of the Workgroup Human-Computer Interaction and*


70. Litwin, M.S. How to measure survey reliability and validity. 1995.


Appendices Summary

Appendix A: Analysis Details
This appendix provides the data that were used as input to the qualitative comparative analysis (QCA). The raw and calibrated data is presented.

Appendix B: Team A Case Study
This appendix provides the case study report that was given to Team A based on assessment using the Team Interaction Framework. The case study includes background on the team and company, a description of the methods used for analysis, findings from the analyses, implications based on the findings, and recommendations for improvements.

Appendix C: Team B Case Study
This appendix provides the case study report that was given to Team B based on assessment using the Team Interaction Framework. The case study includes background on the team and company, a description of the methods used for analysis, findings from the analyses, implications based on the findings, and recommendations for improvements.

Appendix D: QCA Primer
This appendix provides a basic tutorial of how QCA can be used in the field of HCI. It is written in the form of a short conference paper to demonstrate how analysis and findings can be communicated in the publication structure currently used in the HCI community.

Appendix E: Team Interaction Assessment
This appendix provides the questionnaire that was used to facilitate data collection and assessment of the social interaction in the team.

Appendix F: IRB Documentation
This appendix provides the Virginia Tech Institutional Review Board (IRB) approval and informed consent forms required to conduct the study.
Appendix A: Analysis Details

This appendix provides the raw data that were used as input to QCA. The first table shows condition data—before and after calibration for each participant. Similarly, the second table shows the outcomes.

Data Table: Conditions

<table>
<thead>
<tr>
<th>role</th>
<th>sc_s</th>
<th>scs_cal</th>
<th>sc_c</th>
<th>scc_cal</th>
<th>sc_r</th>
<th>scr_cal</th>
<th>ng</th>
<th>ng_cal</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV</td>
<td>0.35</td>
<td>0.01</td>
<td>6</td>
<td>0.5</td>
<td>6.83</td>
<td>0.92</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>UX</td>
<td>0.35</td>
<td>0.01</td>
<td>5</td>
<td>0.05</td>
<td>5.83</td>
<td>0.38</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>0.35</td>
<td>0.01</td>
<td>6</td>
<td>0.5</td>
<td>6.83</td>
<td>0.92</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>UX</td>
<td>0.39</td>
<td>0.04</td>
<td>6</td>
<td>0.5</td>
<td>6.17</td>
<td>0.62</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>6</td>
<td>0.5</td>
<td>6.17</td>
<td>0.62</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>6.5</td>
<td>0.82</td>
<td>5.33</td>
<td>0.12</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>7</td>
<td>0.95</td>
<td>5.5</td>
<td>0.18</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>6.5</td>
<td>0.82</td>
<td>6.17</td>
<td>0.62</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>6.5</td>
<td>0.82</td>
<td>6.67</td>
<td>0.88</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>6</td>
<td>0.5</td>
<td>6</td>
<td>0.5</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>6</td>
<td>0.5</td>
<td>6</td>
<td>0.5</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>7</td>
<td>0.95</td>
<td>6.67</td>
<td>0.88</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>DEV</td>
<td>0.39</td>
<td>0.04</td>
<td>7</td>
<td>0.95</td>
<td>6.67</td>
<td>0.88</td>
<td>b</td>
<td>1</td>
</tr>
</tbody>
</table>

Data Table: Outcomes

<table>
<thead>
<tr>
<th>km_c</th>
<th>kmc_cal</th>
<th>km_e</th>
<th>kme_cal</th>
<th>km_d</th>
<th>kmd_cal</th>
<th>km_u</th>
<th>kmu_cal</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.95</td>
<td>5</td>
<td>0.05</td>
<td>5</td>
<td>0.05</td>
<td>7</td>
<td>0.95</td>
</tr>
<tr>
<td>6</td>
<td>0.95</td>
<td>6</td>
<td>0.5</td>
<td>6</td>
<td>0.5</td>
<td>7</td>
<td>0.95</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>0.95</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>0.95</td>
<td>7</td>
<td>0.95</td>
<td>5</td>
<td>0.05</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>6</td>
<td>0.5</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>6</td>
<td>0.5</td>
<td>5</td>
<td>0.05</td>
<td>5</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>7</td>
<td>0.95</td>
<td>7</td>
<td>0.95</td>
<td>7</td>
<td>0.95</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>6</td>
<td>0.5</td>
<td>7</td>
<td>0.95</td>
<td>6</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>6</td>
<td>0.5</td>
<td>5</td>
<td>0.05</td>
<td>6</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0.05</td>
<td>6</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>6</td>
<td>0.5</td>
<td>6</td>
<td>0.5</td>
<td>6</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Appendix B: Team A Case Study

Team Interaction Case Study: Case A

Jeremy Barksdale
Ph.D. Candidate
(IRB #: 10-180)

Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
336.392.6248
barksdale@vt.edu

Executive Summary

Agile software teams can experience challenges when interacting that may influence how user experience is managed. This case study seeks to understand the social capital, governance form, and user experience knowledge management in one agile software team. Specifically, the focus of this case is on the interaction between developers and user experience designers, but considers insights from other team members as context.

Acknowledgements

I want to thank the team and company for the opportunity to conduct this study: everyone was sincere and helpful. I must specifically thank my contact for her help in coordinating the study with the team and management, and ensuring I had everything I needed to complete the study. I hope the findings are insightful and useful. I realize allowing someone into the company requires trust, and I sincerely value that responsibility. Please don’t hesitate to let me know if I need to clarify anything or if anything in this report appears inaccurate.
# Table of Contents

Executive Summary ................................................................................................................ 1  
Introduction ......................................................................................................................... 3  
Background .......................................................................................................................... 3  
  Company Background ........................................................................................................ 3  
  Project Team Background ................................................................................................. 3  
Agile Project Team Environment and Practices .............................................................. 3  
Development Team Practices .............................................................................................. 4  
User Experience (UX) Team Practices .............................................................................. 4  
Interaction between Development and UX ........................................................................ 4  
Methodology ....................................................................................................................... 5  
  Data Collection ................................................................................................................. 5  
  Data Analysis ...................................................................................................................... 5  
    Social Network Analysis ................................................................................................. 5  
    Statistical Analysis .......................................................................................................... 6  
    Thematic Analysis ........................................................................................................... 6  
Results .................................................................................................................................. 6  
  Social Network Analysis (SNA) ......................................................................................... 6  
  Structural Dimension ........................................................................................................ 6  
    Cognitive Dimension ...................................................................................................... 10  
    Relational Dimension .................................................................................................... 10  
  Network Governance ........................................................................................................ 10  
  Knowledge Management ................................................................................................. 11  
  Thematic Analysis ............................................................................................................ 11  
Discussion ............................................................................................................................ 12  
  Practical Implications and Recommendations ............................................................... 13  
  Limitations of this Research ............................................................................................ 13  
Conclusions .......................................................................................................................... 14  
References ............................................................................................................................ 14
Introduction

Developers and user experience professionals can experience barriers to communicating and collaborating. These challenges can result from a difference in academic background, practical experience, and the like. Although there are a number of approaches to integrating agile and user experience within a team [2], this case study focuses on how the social interactions affect the integration. Specifically, this case considers how the team aligns with the team interaction framework [2] toward understanding how the team’s social capital and governance structure affect how usability knowledge is managed.

Background

Company Background

The company studied for this case is an information technology (IT) services government contractor located just outside the Washington, DC metropolitan area. As an IT services company, they develop software, but do not provide development support. The company is relatively small with one office location. Non-administrative (operational) positions include: software developers, user experience designers, testing staff, documentation staff, community support staff, and product owners.

Project Team Background

Each project team has the flexibility to adopt various practices based on what best fits their needs. Hence, some project teams may be more agile than others. The project team studied for this case is responsible for developing widgets, of old applications, that will run inside of a larger framework. The project team has approximately 50 members. Of those members, there are approximately six developers and one user experience designer. Project team meetings serve primarily as development team meetings, but the UX attends them as well. UX also has separate meetings, which development team members do not typically attend.

Agile Project Team Environment and Practices

The project team uses two week sprints, with no rest after each sprint, and there are six sprints in each release: five development sprints and one stabilization release with regression testing. The developers and UX designers (along with the other project team members) participate in the project standups and retrospectives. During the standups, each member provides a status of what they are working on. If a team member wants to talk about something in detail, the topic is added to the whiteboard during the scrum. Each member can elect to stay for the tailgating session to discuss those items or they can leave. There were framed posters of agile-related content (e.g., manifesto, twelve principles, and rules for scrum) in the rooms where standups occurred. During the retrospectives, project team members discuss...
the pros and cons of the last sprint. These meetings provide a formal setting for the developers and designers to interact.

**Development Team Practices**

The development team holds a planning and costing meeting and use Jira [1] to keep track of their stories, subtasks, resource allocation, and other task related data. The development team uses point cards to estimate how long it will take a developer to implement a story [5]. Each developer holds up a 1, 2, or 3 to denote how long they think the story should take, along with their rationale. This continues until all developers are in consensus. Approximately 20 points are allocated to each sprint. The development team also performs a UX walkthrough to get accustomed to, and provide feedback about, the mockups created by the UX team.

**User Experience (UX) Team Practices**

The UX team holds separate daily standups and weekly retrospective meetings. During standups, each UX team member states what they have been working on or accomplished, what they will be working on that day, and any challenges they may be experiencing. These activities seemed to include anything a UX team member was working on, even if it wasn’t a design or evaluation activity. For example, at least 3 team members mentioned sprint planning as one of their activities. During UX team retrospectives, the team discusses ideas or activities that affect the team as a whole. For example, they discussed creating a UX community/team blog, their interaction with development in terms of honoring the mockup, and practices they would like to adopt such as receiving development’s needs at least one sprint ahead to prevent a rush. The UX team uses kanban [5]—implemented via AgileZen [7]—as part of their practices to keep track of their activities and manage workflow. AgileZen does not replace Jira, but primarily serves to facilitate communication within the UX team.

**Interaction between Development and UX**

The developers and UX designers commonly interacted through meetings (viz., standups, retrospectives, costing and planning), in passing through the halls, via office visits, and electronically (e.g., email). It appeared that most office visits occurred from the UX designer approaching the developer. Interaction also occurs semi-formally through UX brown bags. These brown bags are lunch sessions the UX team created to provide developers an opportunity to gain insight and provide feedback about ideas and designs on which UX team members are working during the conceptualization process.

However, the interaction between development and UX was not always collaborative. Prior to the designers currently employed by the company, the one project member that served as the UX designer was seen as an external resource. Developers and the UX designer would fight, developers wouldn’t wait for the mockup, and developers would not consult UX.
Methodology

Data Collection
Data were collected, and used, from seven members of the project team across varying roles: development, UX, documentation, product owner, and testing via observation, survey, and spot interviews. A background of the company and the teams were obtained on the first day to establish context for the study. Sit-ins with UX or development team members occurred each day. Observations of meetings and one-on-one interactions between developers and UX occurred during the sit-ins.

Data Analysis
Data were analyzed using social network analysis, statistical analysis, and thematic analysis. A social network analysis was used to identify any effects from the social structure of the team. Statistical analysis (viz., Krippendorff’s alpha) was performed to identify levels of agreement among participants as part of the social network analysis. A thematic analysis was used to identify themes from observation and interview data.

Social Network Analysis
The structural dimension of the network—comprised of actors and their ties—was evaluated using the density and degree centralization (via ORA Software [3]) based on responses provided by the entire team about how often they collaborated with, sought help, or sought advice from other members in the team. The density is “the proportion of possible lines that are actually present in the graph” [9]. A network with all lines (or ties) connecting all nodes (or actors) will have a density of 1, and a network with no lines (or ties) connecting any nodes (or actors) will have a density of 0. Density can be used as an indicator of the level of cohesion (or “knittedness” [9]) among members in a network. Although there is no absolute interpretation of density, the result is usually interpreted relative to other relations (e.g., the density of collaboration vs. advice relations). Hence, an “optimal” density is dependent on the relation and other factors specific to the network such as its size.

The degree centralization indicates “how centralized the degree of the set of actors is” [9]. When all other actors choose the same actor, only, for some relation (e.g., with whom they collaborate), the centralization value for the network is 1 (i.e., highly centralized with a star shape). When each actor does not choose, only, the same actor for some relation, the centralization value for the network is 0 (i.e., not centralized with a circle shape).

A compromise between complete- and available-case approaches [8] was used to determine which actors to include when calculating the network’s density and degree centralization. This means that all available participant data were used for project members. Any actors (and respective relations) that are not part of the project team under study were removed since the project was the network boundary. All ‘1’ ratings were excluded from the calculation of the density and centralization since they were explicit
indicators of no interaction (similar to the implicit indication of not expressing a relationship with another team member). Relationships of rating ‘2’ or ‘3’ are represented as dotted lines while relationships of rating ‘4’ or ‘5’ are represented as solid lines in the network diagram.

The relational dimension was measured by calculating agreement among user experience and developer participants’ responses to questions about trust, norms, and values. Agreement of trust was determined by calculating the mean of the likert item responses. Agreement on norms and values was determined by calculating Krippendorff’s alpha for the top three norms and values ranked by developer and user experience participants.

The cognitive dimension was measured by calculating agreement among user experience and developer participants on how familiar members of each role were with the responsibilities of the other role.

Statistical Analysis
Krippendorff’s alpha (instead of Cohens or Fleiss’s kappa) was used to measure level of agreement among participants because there were multiple raters and missing data [6]. Participant responses to questions 13-18 were used to measure agreement of trust among developers and user experience designers. Scores where $\alpha = 1$ indicates perfect reliability, $\alpha = 0$ indicates the absence of reliability (i.e., units and the values assigned to them are statistically unrelated), and $\alpha < 0$ indicates disagreements that are systematic and exceed what can be expected by chance.

Thematic Analysis
Toward identifying themes in the data, a concept map of phrases that linked to a common concept were related and reviewed for consistency with other phrases within the same theme.

Results

Social Network Analysis (SNA)

Structural Dimension
Following are the density and degree centralization results for the collaboration, help, and advice relations. These results reflect how often team members collaborate with each other, and how often they seek help and advice from one another. The network diagrams provide a graphical representation of each relation. The center actor in the graph is the one with the highest betweeness centrality (i.e., the member most present in the shortest path between members, thus, having the potential influence to control the interaction between actors on that shortest path) [9]. It’s important to note that some unidirectional relations are due to the absence of data from those members (i.e., a member with no outgoing lines means no data is available from that member or they do not consider a given relationship to exist).
Density
The collaboration relation reflects how frequently each team member collaborates with each of their remaining teammates. The density of the collaboration relation is 34.6%, and there are 54 connections (or ties) out of a 156 total possible connections. This means that collaboration occurs in 54 directions and, thus, occurs between or among a small subset of the team. As a measure of cohesion, the team has relatively low connectedness in terms of collaboration (as seen in the network diagram).

The network diagram not only illustrates sparse collaboration, but also that there is limited strong and mutual collaboration among team members. For example, there is a weak (dotted) one-way collaboration from a developer to a UX team member indicating that the developer does not often collaborate with the UX member and that the UX member does not consider the relationship with the developer as collaborative.
Centralization

Degree centralization for the collaboration relation is 30.5%, which means that there are few team members with a higher centrality than the other members. Collaboration is generally dispersed (i.e., some level of collaboration occurs among most or all members), but likely occurs at times with a few team members more often than with the rest of the team. These results suggest that the team is not well connected, but that collaboration freely occurs between members.

As indicated in the network diagram, one Documentation member has the highest betweenness. In this case, one might conclude that the shortest path between two actors, for collaboration, is most likely to include the Documentation member than other members.

Help Relation.

<table>
<thead>
<tr>
<th>Density: 35.9%</th>
<th>Freeman's Degree Centralization: 21.3%</th>
</tr>
</thead>
</table>

Density
The help network results reflect how frequently each team member solicits help from each of their other team members. The density of the help relation is 35.9%, and there are 56 connections (or ties) out of a total possible of 156. This means that a small subset of the team seeks help from other members. As a measure of cohesion, the team has relatively low connectedness when it comes to seeking help from one another (as seen in the network diagram).
The network diagram illustrates that help is not often sought within and among the team. For example, there is a dotted line from a developer to a UX team member indicating that help is not often sought from the UX member by the developer, but that the UX member does not seek help from the developer.

Centralization

Degree centralization for the help relation is 21.2%, which means that there are few team members with a higher centrality than the other members. Help is generally dispersed (i.e., help is sought from most or all team members), but likely occurs at times with a few team members more often than with the rest of the team. These results suggest that the team is not well connected, but that help is freely sought and given between members without much need to go through central actors.

As indicated in the network diagram, a Developer member has the high betweenness centrality. In this case, one might conclude that the shortest path between two actors, for collaboration, is most likely to include a Developer member than other members.

Advice Relation.

<table>
<thead>
<tr>
<th>Density: 34.6%</th>
<th>Freeman’s Degree Centralization: 30.5%</th>
</tr>
</thead>
</table>

The advice network results reflect how frequently each team member asks for advice from each of their other team members. The density of the advice relation is 34.6%, and there are 54 connections (or ties) out of a total possible of 156. This means that a small subset of the team seeks advice from other
members. As a measure of cohesion, the team has relatively low connectedness when it comes to seeking advice from one another (as seen in the network).

The network diagram illustrates that advice is not frequently sought within and among the team. For example, there is a dotted line from a developer to a UX team member indicating that help is not often sought from the UX member by the developer, but that the UX member does not seek help from the developer.

Centralization

Degree centralization for the advice relation is 30.5%, which means that there are few team members with a higher centrality than the other members. Advice is generally dispersed (i.e., advice is sought from most or all members), but likely occurs at times with a few team members more often than with the rest of the team. Results suggest that the team is not well connected, but that advice is freely sought and given between members without much need to go through central actors.

As indicated in the network diagram, a Documentation member has the high betweenness centrality. In this case, one might conclude that the shortest path between two actors, for advice, is most likely to include the Documentation member than other members.

In summary, across all three networks: the mean density is 0.350, the mean centralization is 0.274, thus making the structural dimension score 0.312 (i.e., the mean of the total density and total centralization). In general, this project team is not very close (or knitted) or centralized.

Cognitive Dimension

The shared meaning score of 5.75 (out of 7) indicates that participant user experience and developer members between somewhat agree and agree that they are familiar with each other’s responsibilities.

Relational Dimension

The network trust score of 5.92 indicates that the participant user experience and development team members (almost) strongly agree that they can trust their functional and project teammates.

The norms score of .125 indicates that there is low agreement on key norms among participant developers and user experience members.

The values score .727 indicates that there is high agreement on key values among participant developers and user experience members.

Network Governance

The network governance score of .5 indicates that developers and user experience participants are in 50% agreement about what form of governance is used. The majority agreement is that the team...
employs a shared governance form (i.e., that decisions about how the team functions are made by all or most members).

**Knowledge Management**

The knowledge construction score of 5 (out of 7) indicates that development and user experience participants believe that usability ideas are frequently discussed within the team.

The knowledge embodiment score of 4.5 (out of 7) indicates that development and user experience participants believe that usability ideas are embodied into an artifact within the team between sometimes and frequently.

The knowledge dissemination score of 4 (out of 7) indicates that user experience and development participants believe that usability ideas are sometimes disseminated within the team.

The knowledge use score of 6.5 (out of 7) indicates that user experience and development participants believe that usability ideas are incorporated into a build within the team between usually and always (or every time).

What is important to note in these results is that the highest score is with knowledge construction and use. This could likely mean that most constructed knowledge is used without going through the embodiment and dissemination phases. This can occur, for example, when user experience and development are working closely on the product such that they are both at the computer at the time of implementation.

**Thematic Analysis**

Following are the themes from the thematic analysis:

**Conflict between UX and Dev.** This theme reflects conflict, or its potential, between user experience professionals and developers. Participants noted conflict that has existed between UX and development from the time prior to the addition of a UX professional and since that time. Specifically, it was noted that the person that initially served a user experience role just emailed [images] and demanded that the team work based on the images. It was also mentioned that development wouldn’t wait for mockups from user experience and wouldn’t consult UX.

**Interaction Improvement Strategies.** This theme reflects strategies that have been employed toward minimizing or resolving the conflict. As a way to minimize the conflict between UX and development, various improvement strategies were mentioned or observed. These included: employing agile practices in the UX team (e.g., retrospectives and standups), conducting brownbags, and UX communicating “kudos” when they receive positive feedback from customers or users.
UX Review. This theme reflects UX team involvement in the review of the software or ideas. Part of the challenge between UX and development was that the implementation was different from the design. As development and UX started to resolve their conflict, QA became unsure of how to evaluate the UX of the product. As a result, UX now works with QA to ensure the product matches the mockups.

UX Inclusion. This theme reflects the extent to which UX is included in the software development process. There were many opportunities for the inclusion of user experience within the team and, generally, the company. Participants mentioned that there is better communication with the client via UX (versus with development directly), that development and UX consult each other when creating mockups and during implementation, and that UX walkthroughs are part of costing. Also noted was one area where inclusion did not occur: namely, that user experience attends development/project retrospectives, but that development does not attend the UX retro.

Developer Openness to UX. This theme reflects comments or observations about how open development is to UX activities and staff. There were several observations that indicated openness to understanding user experience by development. These include one participant reading through the usability findings on his own accord, a developer listening to a UX designer provide background on visual perception as it related to the developer’s tasks, and development communicating a user experience expectation during a demo.

Discussion

The results suggest that the team’s characteristics—in terms of the social network, network governance form, and management of user experience knowledge—are internally consistent. The low centralization coincides with the shared governance finding. The high agreement on values and shared meaning could facilitate or support the presence of trust within the team. Finally, many user experience ideas may flow from discussion to incorporation into the product (thus, skipping embodiment and dissemination) in a network of trust, low centralization, and high agreement of agile values. The thematic analysis results are also consistent with the other findings, but offer a different perspective: evolution. The results suggest that the team has evolved into its current dynamic through the incorporation of interaction improvement strategies that have contributed to their inclusion and openness to UX by development participant members.

Despite the consistency among most characteristics, there is low agreement on norms. This result is unexpected given the high agreement on values. Among the possible explanations explanation for this inconsistency, is that trust and the high agreement on values affords members flexibility in deviating from or acting independent of conventional norms. It could also be the case that the norms exist, but are not as high of a priority as the values. In this case, team members may not be as familiar with or aware of which norms are most prevalent.
Practical Implications and Recommendations

The findings from this research have implications for how the team interacts. In this section, implications and recommendations for how the findings can be applied toward the team’s interaction are provided. Although recommendations are presented, their ultimate success is dependent on relevance to team goals and their implementation. Here, they are offered as possible courses of action with an assumption that they align with the team’s objectives.

The team could experience difficulty when establishing relational norms. The more cohesive groups are, the more they are influenced by group norms [9]. Hence, minimal cohesion could suggest that there is difficulty in establishing and fulfilling norms within the team. To establish norms in difficult situations, the team can build greater cohesion (i.e., density) by better including team members that have weak or missing ties. Additionally, the team can foster a climate with greater accountability. It would be important, however, to ensure density remains within a moderate range to maintain network efficiency. Too much density could result in coordination problems.

The team may experience challenges with maintaining momentum when collaborating. Collaboration typically entails strong cooperation between people. The presence of one-way and weak collaboration ties could indicate that collaboration among members is not optimal. One possible solution is for team members to leverage the strength of weak ties [4] toward improving collaboration with significantly increasing the density of the network. One example where this seems to already exist is through the role of the Product Owner: the Product Owner not only leads UX, but also plays a critical role in determining development priorities. This puts the Product Owner in a position to advocate for UX and influence the use of user experience ideas.

The team appears to have grown beyond the initial conflict. Shared meaning and governance, trust, and the employment of interaction improvement strategies suggest that the team is effectively moving beyond conflict. As a result, it is possible that user experience ideas are moving from construction to use (thus, bypassing the need for embodiment and dissemination in some cases). The team has the option to employ a lead structure (via the Product Owner) if coordination or participation challenges their ability to achieve their goals.

Limitations of this Research

The primary limitation of this study was the existence of missing data in the social network analysis. When performing a social network analysis, it is difficult to make the same inferences with responses from a subset of the team as would be possible with responses from all members of the team. However, a substantial number of members provided responses such that the input was representative of the whole team.

Cross-sectional studies provide an opportunity to identify team interactions in a short amount of time (like that of taking a snapshot). However, a common limitation associated with cross-sectional studies is
limited exposure to deep experiences or temporal observations that would otherwise be observed (e.g., via an ethnography or longitudinal study). The impact of this limitation is that findings may not account for social interaction that might be observed over a longer term study.

Conclusions

This case study sought to understand how social capital and the way software teams are governed, relate with how user experience knowledge is managed within the team. The team was found to have low density and centralization, high trust, high shared meaning, a shared governance form, and high ratings on knowledge construction and use. Although the team could experience challenges with establishing norms and maintaining momentum during collaboration, they appear to have grown beyond past conflict between user experience and development. Important considerations when moving forward include maintaining a moderate density to minimize possible coordination challenges, leveraging weak ties when improved collaboration is needed, and employing a lead structure when coordination challenges and low participation surfaces to remain effective.

References


4. Granovetter, M.S. The Strength of Weak Ties. American Journal of Sociology 78, 6 (1973), 1360–1380.


Appendix C: Team B Case Study

Team Interaction Case Study: Case B

Jeremy Barksdale
Ph.D. Candidate
(IRB #: 10-180)

Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
336.392.6248
barksdale@vt.edu

Executive Summary

Agile software teams can experience challenges when interacting that may influence how user experience is managed. This case study seeks to understand the social capital, governance form, and user experience knowledge management in one agile software team. Specifically, the focus of this case is on the interaction between developers and user experience designers, but considers insights from other team members as context.

Acknowledgements

I want to thank the team and company for the opportunity to conduct this study: everyone was sincere and helpful. I must specifically thank my contact for his help in coordinating the study with the team and management, and ensuring I had everything I needed to complete the study. I hope the findings are insightful and useful. I realize allowing someone into the company requires trust, and I sincerely value that responsibility. Please don’t hesitate to let me know if I need to clarify anything or if anything in this report appears inaccurate.
**Table of Contents**

Executive Summary .................................................................................................................. 1

Introduction ............................................................................................................................ 3

Background ............................................................................................................................. 3

Company Background ........................................................................................................... 3

Project Team Background .................................................................................................... 3

Agile Project Team Environment and Practices ................................................................. 3

Development Team Practices ............................................................................................... 4

User Experience (UX) Team Practices .................................................................................. 4

Interaction between Development and UX ........................................................................... 5

Methodology .......................................................................................................................... 6

Data Collection ..................................................................................................................... 6

Data Analysis ......................................................................................................................... 6

Social Network Analysis ...................................................................................................... 6

Statistical Analysis ............................................................................................................... 7

Thematic Analysis ................................................................................................................ 7

Results .................................................................................................................................. 7

Social Network Analysis (SNA) ............................................................................................ 7

Structural Dimension .......................................................................................................... 7

Cognitive Dimension ............................................................................................................ 12

Relational Dimension .......................................................................................................... 12

Network Governance .......................................................................................................... 12

Knowledge Management ..................................................................................................... 12

Thematic Analysis ................................................................................................................ 13

Discussion ............................................................................................................................. 14

Practical Implications and Recommendations ................................................................ 14

Limitations of this Research ................................................................................................. 15

Conclusions ......................................................................................................................... 15

References ............................................................................................................................. 15
Introduction

Developers and user experience professionals can experience barriers to communicating and collaborating. These challenges can result from a difference in academic background, practical experience, and the like. Although there are a number of approaches to integrating agile and user experience within a team [1], this case study focuses on how the social interactions affect the integration. Specifically, this case considers how the team aligns with the team interaction framework [1] toward understanding how the team’s social capital and governance structure affect how usability knowledge is managed.

Background

Company Background

The company studied for this case provides enterprise-level hosting services. They are a mid-size software company that develops application programming interfaces (APIs) to their services as well as user-friendly products that consume their APIs. As part of the organizational culture, teams are empowered to establish their own norms and rules as opposed to leadership dictating how the teams should function.

In terms of the product, product managers determine the vision for the product and functions much like a chief executive officer (CEO). Technical product managers are responsible for the day to day management of the product and function much like a chief operating officer (COO).

Project Team Background

The project team, studied, is comprised of Developers (dev), User Experience (UX) designers, a Quality Engineer (QE), a Product Manager (PM), and a Technical Product Manager (TPM). The UX designer and QE are matrixed into the team as needed and are remotely located. Four designers on the team encompass roles from Product Designer (focuses on what a product should do) to Interaction (ix) Designer (focuses on how the product does it).

The project team’s focus is the development of the user-friendly product that consumes relevant service APIs. Each project team has the flexibility to adopt various practices and use various tools based on what best fits their needs. Hence, some project teams may be more agile than others.

Agile Project Team Environment and Practices

The project team follows a continuous integration practice. Developers are generally responsible for writing tests that are automatically executed when submitted, which was traditionally a QE
responsibility. QE s are now responsible for improving the infrastructure of the code base, performance time of the tests, and helping to set up journey tests (across multiple stories).

The overall project is guided by the KanBan [3] process instead of an iterative development process to keep the flow with shorter lead time. With iterative development, development work was getting delayed, which resulted in churn. With KanBan, there is no big deliverable at the end of a sprint because stories are delivered when they are finished. Buffers are areas in the process where bottlenecks might occur and have a specified limit (e.g., 10 stories).

Using KanBan, the project team goes through a series of phases:

1. **Desired Outcome.** End goals for the product are determined through brainstorming among collaborators, which include both UX and development.
2. **UX Phase.** Higher-level mockups (each mockup capturing a handful of stories) are reviewed with external stakeholders. Developers are not heavily involved in this phase.
3. **Start creating epochs** into the backlog.
4. **Analysis Ready** (buffer).
5. **Analysis Phase.** Product managers determine priorities, TPMs work with UX on proposed mockups to assess feasibility and determine acceptance criteria, and TPMs break epochs into stories the developers will implement.
6. **Dev Ready** (buffer).
7. **In Dev Phase.** The TPM manages priorities and the work in progress (WIP) limit as part of ensuring flow.
8. **Dev Done Phase.** The TPM looks to see if everything meets the acceptance criteria. If not, the item moves back to the in dev phase.
9. **Accepted.** Items move into this phase once it passes the acceptance test performed by the TPM.

**Development Team Practices**

The development team primarily follows the practices detailed as environment and practices. In addition to those practices, the development team practices test driven development (TDD) and holds retrospectives (but without the planning task). Developers are co-located and sit near one another in an open space. Most interaction among developers (and the TPM) occurs via formal meetings using meeting rooms or google hangout, talking across the room, and Internet Relay Chat (IRC).

**User Experience (UX) Team Practices**

The UX team also follows agile planning practices. They have two weeks of distinct projects (it’s easier to manage two week sprints and because other development teams are on three week sprints). The UX manager priorities UX work for the month, although not exactly in sync with development. The project-level KanBan process works out better than the iterative approach to development for the UX team because there’s no mad rush at the end of the sprint and because the team has recently added the TPM role.
The UX team has a daily stand-up meeting for approximately an hour to review work-in-process. Each week, the UX team syncs with the TPM and PM to discuss product features that are 80-90% done. The UX manager meets with the PM to discuss long term ideas. Prior to the TPM role, the UX manager and the PM would discuss more details. Even though the PM has the final say on the design and functionality of the product, there is collaboration between the UX manager and the PM. However, sometimes the UX team has to design within the PM’s timescale.

Usability studies are conducted prior to development, approximately every month with 5-10 users, and usually occur if there are a lot of design options and debate within the team. During these studies, the UX designers prototype new interactions, patterns, and products. Testing an application currently doesn’t fit into the engineering backlog and is done about every couple of months.

**Interaction between Development and UX**

Interaction between the UX designers and developers occurs primarily through the TPM. Before the TPM role was created, UX designers were expected to have technical expertise and look through the API so they could label the designs with API specification and calls. Since the TPM role was created, the TPM helps with providing technical insight before starting the project, but has resulted in less interaction between the UX designers and developers. TPMs have a development background and will talk with the devs for something css related.

Synchronization between developers and designers also hasn’t occurred directly because the three hour time differences makes it difficult: there are no more development sprint planning meetings (which UX used to attend), and there are no longer demos showing the product in preproduction to alleviate what was difficult for the UX team to see remotely. UX and development also have (separate) IRC channels for chatting.

When the UX and development teams were co-located, frustration was exposed during conversations as one role might not want to bend. When remote, there isn’t as much emotional awareness and influence. Often, UX and dev are not brought into each other’s conversations. Sometimes, a person will frequent someone they know well (e.g., if a developer has a personal relationship with a UX designer), but that developer or designer may not have been the one to work on the problem being discussed or the person could become burdened from the frequent contact. It also happens that side conversations ripple throughout the application, without team review, because of a quick decision made between a developer and designer.
Methodology

Data Collection
Data were collected via observation, survey, and as-needed interviews. A background of the company and the teams were obtained on the first day to establish context for the study. Sit-ins with development occurred each day. Observations of meetings and one-on-one interactions between developers and UX occurred as part of the sit-ins.

Data were collected, and used, from 13 members of the project team across varying roles: development, UX, technical product management, and quality engineering.

Data Analysis
Data were analyzed using social network analysis, statistical analysis, and thematic analysis. A social network analysis was used to identify any effects from the social structure of the team. Statistical analysis (viz., Krippendorff’s alpha) was performed to identify levels of agreement among participants as part of the social network analysis. A thematic analysis was used to identify themes from observation and interview data.

Social Network Analysis
The structural dimension of the network—comprised of actors and their ties—was evaluated using the density and degree centralization (via ORA Software [2]) based on responses provided by the entire team about how often they collaborated with, sought help, or sought advice from other members in the team. The density is “the proportion of possible lines that are actually present in the graph” [W&F, pg. 101]. A network with all lines (or ties) connecting all nodes (or actors) will have a density of 1, and a network with no lines (or ties) connecting any nodes (or actors) will have a density of 0. Density can be used as an indicator of the level of cohesion (or “knittedness” [6]) among members in a network. Although there is no absolute interpretation of density, the result is usually interpreted relative to other relations (e.g., the density of collaboration vs. advice relations). Hence, an “optimal” density is dependent on the relation and other factors specific to the network such as its size.

The degree centralization indicates “how centralized the degree of the set of actors is” [6]. When all other actors choose the same actor, only, for some relation (e.g., with whom they collaborate), the centralization value for the network is 1 (i.e., highly centralized with a star shape). When each actor does not choose, only, the same actor for some relation, the centralization value for the network is 0 (i.e., not centralized with a circle shape).

A compromise between complete- and available-case approaches [5] was used to determine which actors to include when calculating the network’s density and degree centralization. This means that all available participant data were used for project members. Any actors (and respective relations) that are
not part of the project team under study were removed since the project was the network boundary. All ‘1’ ratings were excluded from the calculation of the density and centralization since they were explicit indications of no interaction (similar to the implicit indication of not expressing a relationship with another team member). Relationships of rating ‘2’ or ‘3’ are represented as dotted lines while relationships of rating ‘4’ or ‘5’ are represented as solid lines in the network diagram.

The relational dimension was measured by calculating agreement among user experience and developer participants’ responses to questions about trust, norms, and values. Agreement of trust was determined by calculating the mean of the likert item responses. Agreement on norms and values was determined by calculating Krippendorff’s alpha for the top three norms and values ranked by developer and user experience participants.

The cognitive dimension was measured by calculating agreement among user experience and developer participants on how familiar members of each role were with the responsibilities of the other role.

Statistical Analysis
Krippendorff’s alpha (instead of Cohen’s or Fleiss’s kappa) was used to measure level of agreement among participants because there were multiple raters and missing data [4]. Participant responses to questions 13-18 were used to measure agreement of trust among developers and user experience designers. Scores where $\alpha = 1$ indicates perfect reliability, $\alpha = 0$ indicates the absence of reliability (i.e., units and the values assigned to them are statistically unrelated), and $\alpha < 0$ indicates disagreements that are systematic and exceed what can be expected by chance.

Thematic Analysis
Toward identifying themes in the data, a concept map of phrases that linked to a common concept were related and reviewed for consistency with other phrases within the same theme.

Results
Data were collected from # of participants, types of participants, and other descriptives

Social Network Analysis (SNA)

Structural Dimension
Following are the density and degree centralization results for the collaboration, help, and advice relations. These results reflect how often team members collaborate with each other, and how often they seek help and advice from one another. The network diagrams provide a graphical representation of each relation. The center actor in the graph is the one with the highest betweenness centrality (i.e., the member most present in the shortest path between members, thus, having the potential influence to control the interaction between actors on that shortest path) [6]. It’s important to note that some
unidirectional relations are due to the absence of data from those members (i.e., a member with no outgoing lines means no data is available from that member or they do not consider a given relationship to exist).

**Collaboration Relation.**

<table>
<thead>
<tr>
<th>Density: 40%</th>
<th>Freeman’s Degree Centralization: 26.6%</th>
</tr>
</thead>
</table>

![](image)

*Density*

The collaboration relation reflects how frequently each team member collaborates with each of their remaining teammates. The density of the collaboration relation is 40%, and there are 112 connections (or ties) out of 280 total possible connections. This means that collaboration occurs in 112 directions and, thus, occurs between or among a moderate subset of the team. As a measure of cohesion, the team has relatively moderate connectedness in terms of collaborating (as seen in the network diagram).

The network diagram illustrates that there is moderate collaboration, but also that there is limited strong collaboration among team members. For example, most collaboration ties with UX members are
weak (dotted) indicating that collaboration does not occur often with the UX members. However, it’s important to note that some unidirectional relations are due to the absence of data from those members (i.e., a member with no outgoing lines means no data is available from that member or they do not consider a given relationship to exist).

Centralization

Degree centralization for the collaboration relation is 26.6%, which means that there are few team members with a higher centrality than the other members. Collaboration is generally dispersed (i.e., some level of collaboration occurs among most or all members), but likely occurs at times with a few team members more often than with the rest of the team.

As indicated in the network diagram, one Development member has the highest betweenness. In this case, one might conclude that the shortest path between two actors, for collaboration, is most likely to include the Development member than other members.

Help Relation.

| Density: 40% | Freeman’s Degree Centralization: 27.7% |

---

Page 9 of 16
Density

The density of the help relation is 40%, and there are 112 connections (or ties) out of a total possible of 280. This means that a moderate subset of the team seeks help from other members. As a measure of cohesion, the team has relatively moderate connectedness when it comes to seeking help from one another (as illustrated in the network diagram).

It’s important to note that some unidirectional relations are due to the absence of data from those members (i.e., a member with no outgoing lines means no data is available from that member or they do not consider a given relationship to exist).

Centralization

Degree centralization for the help relation is 27.7%, which means that there are few team members with a higher centrality than the other members. Help is generally dispersed (i.e., help is sought from most or all team members), but likely occurs at times with a few team members more often than with the rest of the team.

As indicated in the network diagram, a Developer member has the high betweenness centrality. In this case, one might conclude that the shortest path between two actors, for collaboration, is most likely to include a Developer member than other members.
Advice Relation.

Density: 36.1%  
Freeman’s Degree Centralization: 33.7%

Density
The density of the advice relation is 36.1%, and there are 101 connections (or ties) out of a total possible of 280. This means that a small subset of the team seeks advice from other members. As a measure of cohesion, the team has relatively low connectedness when it comes to seeking advice from one another (as seen in the network diagram).

It’s important to note that some unidirectional relations are due to the absence of data from those members (i.e., a member with no outgoing lines means no data is available from that member or they do not consider a given relationship to exist).

Centralization
Degree centralization for the advice relation is 33.7%, which means that there are few team members with a higher centrality than the other members. Advice is generally dispersed (i.e., advice is sought
from most or all members), but likely occurs at times with a few team members more often than with the rest of the team.

As indicated in the network diagram, a TPM member has the high betweenness centrality. In this case, one might conclude that the shortest path between two actors, for collaboration, is most likely to include a TPM member than other members.

Cognitive Dimension
The shared meaning score of 6.36 indicates that participant user experience and developer team members between agree and strongly agree that they are familiar with each other’s responsibilities.

Relational Dimension
The network trust score of 6.12 indicates that the user experience and development participant team members agree that they can trust their functional and project teammates.

The norms score of -.38 indicates that there is systematic disagreement of norms among participant development and user experience team members.

The values score of .097 indicates that there is low agreement on values among the participant developer and user experience team members.

Network Governance
The network governance score indicates that developers and user experience participants are in 49% agreement about what form of governance is used. The majority agreement is that the team employs a lead governance form (i.e., that decisions about how the team functions are made by one or a select few members).

Knowledge Management
The knowledge management construction score of 4.1 (out of 7) indicates that development and user experience participants believe that usability ideas are sometimes discussed within the team.

The knowledge management embodiment score of 5.2 (out of 7) indicates that development and user experience participants believe that usability ideas are slightly more than frequently embodied into an artifact within the team.

The knowledge management dissemination score of 4.8 (out of 7) indicates that development and user experience participants believe that usability ideas are (almost) frequently disseminated within the team.
The knowledge management use score of 4.8 (out of 7) indicates that development and user experience participants believe that usability ideas are (almost) frequently incorporated in a build within the team.

**Thematic Analysis**

Following are the themes from the thematic analysis:

**Conflict between UX and Dev.** This theme reflects conflict, or its potential, between user experience professionals and developers. Participants noted areas of (potential) conflict between UX and development. Specifically, they mentioned that in person communication is more difficult than via chat, being remote means not knowing what is being worked on or who is working on what, and that you don’t know what’s going on since you can’t see the emotions remotely (although colocation exposes frustrations).

**Interaction Improvement Strategies.** This theme reflects strategies that have been employed toward minimizing or resolving the conflict. The primary strategy for improving the interaction between UX and development was the creation of the TPM role. As a result, UX is no longer expected to have the technical expertise required to label designs with API specs. Other strategies include the UX designers visiting the geographic location of the developers to establish personal relationships and work with development directly, and a UX/developer frequenting a developer/UX they know well.

**UX Review.** This theme reflects UX team involvement in the review of the software or ideas. Based on observation, the user experience of the product is reviewed by the TPM as part of acceptance testing. UX does not appear to have an active role in ensuring the product deliverable matches the mockups.

**UX Inclusion.** This theme reflects the extent to which UX is included in the software development process. UX inclusion seemed minimal based on the observation and interview data. Participants noted that there was not much interaction between UX and development during story development, UX does not attend retrospectives (although they did attend development demos, which are ending), development and UX use separate chat channels, and UX and development aren’t brought into each other’s conversations. However, there were some moments of inclusion such as when development and UX brainstorm together at “desired outcome” meetings, during the UX phase of the Kanban development process, during the analysis phase of the Kanban process via the TPM, and when usability studies are conducted prior to development.

**Developer Openness to UX.** This theme reflects comments or observations about how open development is to UX activities and staff. There were few indications of developer openness to user experience. Participants mentioned that development will sometimes get on the UX chat to request a call, developer demos (to UX) no longer occur, and development is not heavily involved in the UX phase of the Kanban process.
Discussion

The majority agreement among participants was that a lead governance form was employed, and the average centralization of all structural relations (i.e., collaboration, help, and advice) is relatively low (viz., 30%). These results suggest that the team seems to remain relatively decentralized, despite being perceived as having a lead governance form. This result further suggests that members are still encouraged to communicate with each other for collaboration, help, and advice.

The limited interaction between UX and development appear to support the relatively moderate construction, embodiment, dissemination, and use of user experience knowledge. It is likely that the TPM role contributes to the moderate (i.e., not low) user experience knowledge management ratings. However, it appears that there is the opportunity for more user experience knowledge construction, embodiment, dissemination, and use.

The result of knowledge management shows that user experience knowledge is embodied more frequently than it is disseminated and used. This result indicates that user experience ideas that are embodied into artifacts (e.g., wireframes, mockups, etc.) are not frequently disseminated to the team and incorporated into the product. It’s unlikely that 100% of user experience ideas are used in any company’s product due to varying factors: schedule limitations, feasibility, and so on. Dissemination of user experience ideas could likely result from a limited interaction between development and UX.

Practical Implications and Recommendations

The findings from this research have implications for how the team interacts. In this section, implications and recommendations for how the findings can be applied toward the team’s interaction are provided. Although recommendations are presented, their ultimate success is dependent on relevance to team goals and their implementation. Here, they are offered as possible courses of action with an assumption that they align with the team’s objectives.

The team appears to fundamentally disagree on norms and values. These relational elements are important for resolving conflict and maintaining performance. Given the team acknowledges working in an agile environment where values are an explicit component, the low agreement on values could indicate that UX and development are not communicating about, and reaching agreement on, which values are most important. Identifying and agreeing on the most important norms and values could help development and user experience lead to more inclusion and sharing between development and user experience.

Despite the moderate level of cohesion (i.e., density) among team members, the thematic analysis results suggest that the interaction between user experience and development may not be as cohesive as the entire team. User experience participants noted the strategies for improving the interaction between developers and themselves. It will be important for other team members to contribute to the improved interaction between the two by such acts as encouraging them to communicate directly with
each other as much as possible, ensuring they are included in each other’s discussions, and facilitating maximal awareness of what each other are doing. In doing so, they entire team can benefit through better alignment of expectations and a more cohesive team.

Limitations of this Research

The primary limitation of this study was the existence of missing data in the social network analysis. When performing a social network analysis, it is difficult to make the same inferences with responses from a subset of the team as would be possible with responses from all members of the team. However, a substantial number of members provided responses such that the input was representative of the whole team.

Cross-sectional studies provide an opportunity to identify team interactions in a short amount of time (like that of taking a snapshot). However, a common limitation associated with cross-sectional studies is limited exposure to deep experiences or temporal observations that would otherwise be observed (e.g., via an ethnography or longitudinal study). The impact of this limitation is that findings may not account for social interaction that might be observed over a longer term study.

Conclusions

This case study sought to understand how social capital and the way software teams are governed, relate with how user experience knowledge is managed within the team. Key findings were that the team has moderate density and centralization, high trust, high shared meaning, a lead governance form, and highest rating for user experience knowledge embodiment. Moving forward, it is important for developers and designers to identify and agree on key values, for designers and developers to include each other in their work activities, and for the project teams to support the improved interaction between developers and designers.

References


Appendix D: QCA Primer

The following is a tutorial for Qualitative Comparative Analysis (QCA). It is structured as a paper to highlight the important aspects of QCA: namely, its background, a type of HCI-related problem it is well-suited to address, and how results are interpreted and reported. A more step-by-step tutorial can be found in books on QCA [107,108].

Introduction

As many experimenters can attest, it can be difficult to obtain enough participants to make inferences using statistical techniques in technology design. Emerging approaches like Qualitative Comparative Analysis (QCA) employ Boolean algebra and set theory to provide an analytic technique that does not require as many samples to make causal inferences [108]. Boolean algebraic approaches to evaluation—commonly used in electronic circuit design—afford identifying causality through performing Boolean operations on data and establishing necessity and sufficiency when paired with set theory. Because statistical methods are based in linear algebra and correlational connections, hundreds of cases (e.g., participants) can be required to fully leverage the power of these approaches. In many situations, QCA can allow interface designers to draw more meaningful conclusions with limited samples.

Qualitative Comparative Analysis

In QCA, as is also true for statistical techniques, the unit of analysis defines a case. Each case contains a set of causal conditions (i.e., independent variables) that contribute to the possible explanation of an outcome (i.e., dependent variable). The combination of conditions for a given case is known as a configuration. Case and configuration selection by researchers is informed by theoretical or substantive knowledge of the domain. Truth table analysis of all possible configurations determines which causal conditions are necessary and/or sufficient to the presence and absence of the outcome. Boolean operations are then performed on causal combinations to identify parsimonious solutions (or model) that best explain the phenomena being studied.

Three variants of QCA provide flexibility in identifying deterministic sets for continuous variables: crisp-set (csQCA), multi-value (mvQCA), and fuzzy-set (fsQCA). These methods balance the strengths of case-oriented (viz., qualitative) and variable-oriented (viz., quantitative). In csQCA, values are dichotomized—phenomena are deemed either present or absent. In mvQCA, values are categorical—a variable can have
many levels. In fsQCA, values are represented in decimal form between 0 and 1—allowing researchers greater precision in their explanation. A combined use of fsQCA and csQCA are the most commonly used forms of QCA, and they were used in this dissertation research.

**Consistency/Coverage and Necessity/Sufficiency**

In statistics, significance and strength play a critical role in interpreting the meaningfulness of results. Consistency and coverage, respectively, play a similar role in QCA [107].

*Consistency* measures how often a given combination of conditions is associated with a specific outcome (i.e., frequency of the combination of conditions), and it is used to determine which configurations deserve further attention. As a heuristic, 85% consistency is a reasonably high enough threshold to indicate that further analysis is warranted. *Coverage* measures the extent to which a solution explains the data. For example, one solution might explain 80% of the cases. Coverage is calculated if consistency is above the researcher’s threshold. When multiple configurations lead to the same outcome (known as equifinality), coverage can indicate which path is more important.

Conclusions are expressed in terms of whether conditions are necessary and/or sufficient to influence the outcome [108]. A condition is *necessary* to cause the outcome if the outcome consistency score is less than or equal to the consistency score of the condition(s). A necessary condition must be present for the outcome to occur. A condition is *sufficient* to influence the outcome if the condition consistency score is less than or equal to the consistency score of the outcome, A sufficient condition can, alone, influence the outcome.

**Comparison of QCA to Statistical Techniques**

A key benefit to using QCA over statistical techniques is that QCA enables researchers to make stronger causal inferences for small-N (fewer than 15 cases) or intermediate-N (between 15 and 100 cases) problems [108]. This is a relatively small requirement, compared to statistics, and is enabled by a Boolean (vs. statistics linear) approach to analysis. QCA also enables the researcher to provide an explanation of phenomena based on multiple conjunctural causation, which views causality as context and conjuncture specific [108]. Thus, instead of isolating variables that influence the outcome (as is the case in statistical techniques), QCA acknowledges that—and seeks to identify—the combination of conditions that influence the outcome. For example, when considering how space between virtual keys, the screen resolution, and screen size effect user satisfaction with a tablet PC, statistical methods might attempt to
isolate the most contributing factor(s), whereas QCA would not only uncover isolating conditions, but
would also seek to provide the combination of conditions that explain the user’s satisfaction.

However, statistical techniques can complement QCA. Some statistical techniques can inform decisions in
QCA, such as selecting conditions based on statistical significance criteria [108]. Statistics can also be used
as a way of setting thresholds when dichotomizing data.

**Example Application**

In collaborative work environments, more lines of communication come with higher transaction costs.
Prior research notes the lower cost associated with sparse networks, acknowledging that sparse networks
could result in less knowledge transfer due to reduced redundancy [72]. Given the inherent interest for
simplicity in agile teams, we consider how usability experts can be incorporated without significantly
increasing the communication overhead. Studies have considered the integration between developers
and usability experts in an agile setting [65], as well as the role of the client in an agile development team.
However, few, if any, have explored the social relationship among the usability expert (e.g., a designer,
usability engineer, etc.), client, and software developers in an agile setting. In our study, we examined the
agile usability team as a social network and evaluated a small team along the structural dimension (viz.,
the impersonal configuration of linkages between actors such as ties) of a previously identified social
capital construct [87]. The communication linkages between the usability expert, developers, and the
client define a team interaction pattern in this study.

The team was tasked with developing a multi-touch education table that would allow elementary school
children to play collaborative games using interaction gestures. The client’s objective was to study how
children used the table to understand how multi-touch technology can be used to influence learning.
Hence, the client’s users were elementary education children, which were not a part of our study of team
interaction patterns.

**Method and Approach**

We employed a comparative case design to learn which communication linkages were necessary and/or
sufficient to achieve high usability knowledge utilization. A case was defined as an interaction pattern
among team members. Three team interaction (Ix) patterns, issued to the team weekly, were used: Ix
pattern A, where the usability expert only interacted with the developers; Ix pattern B, where the usability
expert only interacted with the client; Ix pattern C, where the usability expert interacted with the entire
team. Random investigator visits and inquiry ensured compliance with the interaction pattern of the week.

Participants were a newly formed agile usability team of two software developers, one usability expert, and a client that met face-to-face with the developers and usability expert at least weekly. Every team member except the client worked in the same building. Each participant was provided an overview of the study and relevant agile usability training prior to beginning the project and study. Data were collected after each iteration via a weekly administered online questionnaire for five weeks. Only four weeks of data were used as iteration two was discarded since the client did not provide feedback for that week. The interaction pattern from week two was reissued in a later week to prevent learning effects.

Usability knowledge utilization was measured via self-reported responses from the participants. The conditions were measured as the presence or absence of interaction within the team. No threshold setting was necessary as the state of communication links is inherently binary. Usability knowledge was considered utilized if the usability recommendations were implemented in part or in full, and not utilized if the recommendations were not implemented in the product. A weekly survey contained six questions: one prompting participants to specify their role on the project, one 5-point Likert question to assess how often usability recommendations were used (never/always), and four qualitative questions for a deeper understanding of usability decisions and the team’s interaction.

Analysis

Data were analyzed using fsQCA to support greater precision. Each level of the 5-point response scale for the utilization frequency item was assigned a fuzzy value equally distributed between 0 (never) and 1 (always). Likert responses within an iteration were averaged across participants by taking the mean of all participant responses. The outcome was calibrated according to the desired thresholds of 0.75, 0.5, and 0.25 \(^{107}\). Thresholds were determined in alignment with the likert response scale and the desired level of usability knowledge utilization. In essence, .75 represents that usability knowledge was utilized at least almost every time, .25 represents that usability knowledge was almost never utilized, and .5 represents the crossover point between the desired and undesired utilization level. Hence, responses greater than or equal to 0.75 were categorized as high utilization. Responses at 0.5 were not considered as high or low utilization. Lastly, responses less than or equal to 0.25 were categorized as low utilization.
Standard analyses were then run using the fsQCA software tool, which executes the truth table algorithm [107]. In the resulting truth table, rows without cases are removed. Outcomes were coded as 1 if their consistency score was above the recommended threshold [107] of 0.75, and 0 otherwise.

Table 12: Final truth table before running the standard analysis

<table>
<thead>
<tr>
<th>Pattern B</th>
<th>Pattern A</th>
<th>Number</th>
<th>UK Util.</th>
<th>Raw Consist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1 (33%)</td>
<td>1</td>
<td>.875</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1 (66%)</td>
<td>0</td>
<td>.438</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1 (100%)</td>
<td>0</td>
<td>.688</td>
</tr>
</tbody>
</table>

The final truth table (Table 1) shows the frequency of cases for each unique configuration, and it is used as the basis for the standard analyses. Consistency measures the degree to which the causal combinations are a subset of the set theoretic relation, which will be compared to the outcome consistency score to determine whether necessity and/or sufficiency exist between the conditions and outcomes.

The following steps were taken to perform the analyses:

1. Each Likert item response was mapped to a fuzzy value equally distributed between 0 (never) and 1 (always) based on the Likert response scale.
2. Each case (or row in the data table) was mapped to an interaction pattern and the participant responses were averaged for each case using the mean (calculated from the previous step).
3. The thresholds were calibrated in alignment with response scale to denote the desired level of usability knowledge utilization.
4. The truth table was generated from the raw data table and refined based on the consistency scores of the conditions.
5. The standard analyses algorithm was executed using the refined truth table as input.
6. Post analysis was conducted to determine whether any conditions (or combinations thereof) were necessary and/or sufficient for high usability knowledge utilization.
7. Further analysis was conducted to also determine whether any conditions (or combinations thereof) were necessary and/or sufficient for low usability knowledge utilization (indicated using the ~ symbol for low).

Results and Discussion

Results showed that neither Pattern A nor B, alone, is a necessary condition for usability knowledge utilization given their consistency score is lower than the recommended .90 threshold for necessary
conditions [107], with user_client_ix having a consistency score of .56, and user_dev_ix having a consistency score of .78.

Results also showed that Ix pattern A is sufficient to produce high usability knowledge utilization given its consistency score of .88 (coverage = .44), which is higher than the recommended .80 threshold for sufficient conditions [107]. Hence, the intermediate solution is that high interaction between usability experts and developers along with low interaction between usability experts and clients is sufficient for high usability knowledge utilization:

\[ uku = user\_dev\_ix \land \neg user\_client\_ix \]

Results from analyzing which conditions were associated with low usability knowledge utilization, the consistency scores of the conditions in the truth table were not high enough (i.e., their consistency score was less than .80) for set membership to justify running the standard analyses. Hence, the causal conditions do not contribute to low usability utilization. This is consistent with the finding that neither individual causal condition is necessary.

We found that not all team members have to communicate with the client in order to realize high usability knowledge utilization. This is important considering the need for flexibility and decreased overhead in agile environments.

However, there are possible alternate explanations. Other factors, such as the priority and feasibility of the usability recommendation, can influence whether a usability recommendation is utilized. For example, redesigning the user interface might not be feasible within the current release. Although these are important considerations, this example served as a means to demonstrate, via a simple example, how QCA could be used in interaction design. More studies across projects, teams, organizations, and scenarios are needed to generalize the results.

Discussion

Benefits and costs of QCA
QCA has a number of benefits beyond those already mentioned: the operations are relatively few and simple, the analytical tools are freely available, and it is growing in use. Boolean and set theory techniques
are relatively simple in comparison to statistical methods, and arguably, align more with how we naturally think. Boolean's basic operations are limited to AND (viz., conjunction or set intersection), OR (viz., disjunction or set union) and NOT (viz., negation) and are easy to compute using a truth table. Set theory operations are relatively simple as well, in comparison to probability methods, and their operations compliment Boolean operations.

QCA, developed in the late 1980s for use in comparative politics to compare nation-states, provided an alternative for investigators limited by the finite nature of statistical methods. Since this time, QCA has been applied in other disciplines such as management, forestry, and engineering. As more disciplines adopt QCA, it becomes more robust and rigorous while maintaining relatively low complexity compared to statistical techniques. A growing set of software tools are freely available for csQCA, mvQCA, and fsQCA; specifically, the fsQCA tool (for csQCA and fsQCA), Tosmana (for csQCA and mvQCA) and the statistical software package R (for QCA).

There are costs associated with using QCA: it requires defending against statistical methods, techniques are still maturing to meet researcher needs, and support for learning QCA is minimal. Statistics has become a dominant method in the scientific community as the rigorous approach to conducting analyses. However, as this paper argues, statistics has fundamental and practical limitations, particularly for a field like interaction design. The cost, unfortunately, for employing less-known methods such as QCA, is that its use requires defense to a greater degree than when using statistical methods—and in some cases, due to the mere reluctance to welcome alternatives.

The relative novelty of QCA also means that some of the techniques to accomplish tasks may not be readily available. For example, when QCA first arrived, there was only csQCA, where data had to be dichotomized to conduct analyses. However, since that time, mvQCA and fsQCA have been developed to accommodate the need for greater granularity in the data and precision in the analyses.

Finally, resources for learning QCA are limited in comparison to the plethora of materials, courses, and statisticians available for learning statistical methods. Ragin and others have published books and articles, but research and application are still needed to build a knowledgebase to better facilitate learning and using QCA more fluently.
Implications for interface designers

QCA is advantageous to technology designers as sample sizes are often too small to make causal inferences using statistical reasoning. QCA enables the asking of practical questions where it may be more insightful to understand causal complexity versus an isolation effect. For example, understanding what configuration of user interface elements leads to (i.e., are necessary and/or sufficient for) high task performance might be more insightful than isolating each element’s influence on the task.

Interface designers often collect rich data that can add deeper meaning to the conclusions drawn using QCA compared to statistical techniques. Data is separated from its context for statistical analyses. QCA provides the potential for interaction designers to make rich causal inferences. Putting the time and effort toward testing and adopting QCA in their community—even alongside statistical methods—can start the process of learning the ways in which QCA can enrich our community.

Conclusion

QCA affords performing formal and rigorous analyses that would otherwise 1) yield weaker inferences using statistical methods due to resource limitations, or 2) come under attack from using qualitative methods for lacking the formality and rigor often desired in scientific investigation.
Appendix E: Team Interaction Assessment

This appendix provides the instrument that was used to collect data from participants. The purpose of the instrument was to obtain background about the team and participants, and assess each participant’s perception of the social capital within in the team, governance used in the team, and how knowledge is managed within the team.

INSTRUCTIONS:
Thank you for participating to help us better understand the social interaction in your team. For the first part of the survey, please answer the questions in the questionnaire. During the second part, I will ask you a few follow-up questions. Please be candid in your responses, and let me know if clarification is desired or if you would like to stop at any time.

DEFINITIONS:
Functional Team: The team with whom you share a functional role or expertise (e.g., usability, QA, development, etc.)

Primary Project Team: The multidisciplinary team with whom you share project or product responsibilities (e.g., feature team)

Norms & Values: Values are related to the norms of a culture, but they are more global and abstract than norms. Norms are rules for behavior in specific situations, while values identify what should be judged as good or evil. For example, flying the national flag on a holiday is a norm, but it reflects the value of patriotism [REF: Seyyedi et. al.].

PART I: Questionnaire

Title (e.g., Manager, Director, VP, etc.): _____________________________________________

Primary Role (e.g., Usability expert, Developer, QA, etc.): _______________________________
Years of experience in primary role: 0-4 yrs. _ 5-9 yrs. _ 10-14 yrs. _ 15-19 yrs. _ 20+ yrs.

Length of service with your current organization (in years and months): ______________________

1 In what iteration/sprint is your **primary project**? ______

2 In what release is your **primary project**? ______

3 How many members on your **primary project team** are in your functional team (including yourself)? ______

4 How many years and months have you been on your **primary project**? ______

5 I am satisfied with my job.

   1 Strongly Disagree  2 Disagree  3 Somewhat Disagree  4 Neither Agree or Disagree  5 Somewhat Agree  6 Agree  7 Strongly Agree

6 My overall job satisfaction has changed in the following direction within the last year.

   1 Strongly Decreased  2 Decreased  3 Somewhat Decreased  4 Neither Increased nor Decreased  5 Somewhat Increased  6 Increased  7 Strongly Increased

7 I am satisfied with my **functional team**.

   1 2 3 4 5 6 7
8. I am satisfied with my **primary project team**.

9. I engage in meaningful work.
Please list the names (first name and last initial) and the role of each person that works on your primary project (including yourself). Names provided will remain anonymous and your response will be kept confidential. Use the additional sheet if more space is needed.

a  For each person, indicate how frequently you contact them (for all purposes stated in the respective header) by circling a rating on the given scale for each purpose.

To collaborate: Collaboration involves approaching a team member with the intention of working closely together to achieve a goal.

For help: Help involves approaching a team member to obtain clarification or work together (but not necessarily closely) on a particular issue.

For advice: Advice entails approaching a team member to ask for general guidance or guidance about a particular issue.

Scale: 1=Never, 2=Rarely, 3=Sometimes, 4=Often, 5=Always

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Initial</th>
<th>Role</th>
<th>To collaborate</th>
<th>For help</th>
<th>For advice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>
11 I am familiar with the responsibilities of team members on my primary project team (e.g., feature team or product team) that are outside of my direct responsibility.

12 I am familiar with the responsibilities of team members on my functional team (e.g., usability or development) that are outside of my direct responsibility.

13 I believe that most people can be trusted professionally.

14 I believe that you can’t be too careful in dealing with people professionally.

15 I trust the members on my functional team professionally.
16 I trust the members on my primary project team professionally.

1
Strongly Disagree
2
Disagree
3
Somewhat Disagree
4
Neither Agree or Disagree
5
Somewhat Agree
6
Agree
7
Strongly Agree

17 The members on my functional team are honest.

1
Strongly Disagree
2
Disagree
3
Somewhat Disagree
4
Neither Agree or Disagree
5
Somewhat Agree
6
Agree
7
Strongly Agree

18 The members on my primary project team are honest.

1
Strongly Disagree
2
Disagree
3
Somewhat Disagree
4
Neither Agree or Disagree
5
Somewhat Agree
6
Agree
7
Strongly Agree
Please indicate what norms (i.e., unwritten expectations) and values exist on your **primary project team** (see definition for guidance) by ranking, at least, the top three norms and values. You may write in additional norms and values if the norm or values is not listed. Use the back of the sheet if more space is needed.

<table>
<thead>
<tr>
<th>Norms</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>___ Respect for the worth of every person</td>
<td>___ Individuals and their interactions</td>
</tr>
<tr>
<td>___ Acknowledge that the user is not like you</td>
<td>___ End user collaboration</td>
</tr>
<tr>
<td>___ Truth in communication</td>
<td>___ Customer satisfaction</td>
</tr>
<tr>
<td>___ Welcome changing requirements</td>
<td>___ Simplicity</td>
</tr>
<tr>
<td>___ Transparency of all data, actions, and decisions</td>
<td>___ Customer collaboration</td>
</tr>
<tr>
<td>___ Trust that each person will support the team</td>
<td>___ User satisfaction</td>
</tr>
<tr>
<td>___ Continuously perform usability evaluations</td>
<td>___ Respond to change</td>
</tr>
<tr>
<td>___ Commitment to the team and to the team’s goals</td>
<td>___ Face-to-face conversation</td>
</tr>
<tr>
<td>___ As a team, reflect on how to become more effective.</td>
<td>___ Working software</td>
</tr>
<tr>
<td>___ Keep users informed and in control</td>
<td></td>
</tr>
</tbody>
</table>
20 Generally, who makes most of the decisions about how your primary project team will function day-to-day?
   a. All members
   b. One or a select few members
   c. An external person or group (e.g., Manager, Project Management Office, etc.)

21 How often are usability ideas discussed (e.g., in a meeting, face to face, etc.)?

   1 2 3 4 5 6 7
   Never Rarely Occasionally Sometimes Frequently Usually Every time

22 How often are usability ideas embodied into an artifact (e.g., minutes, email, design, etc.)?

   1 2 3 4 5 6 7
   Never Rarely Occasionally Sometimes Frequently Usually Every time

23 How often are usability ideas disseminated (e.g., to team members, etc.)?

   1 2 3 4 5 6 7
   Never Rarely Occasionally Sometimes Frequently Usually Every time

24 How often are usability ideas incorporated into a build/release?

   1 2 3 4 5 6 7
   Never Rarely Occasionally Sometimes Frequently Usually Every time
Appendix F: IRB Documentation

MEMORANDUM
DATE: January 30, 2013
TO: Scott McCrickard, Jeremy Totton Barksdale
FROM: Virginia Tech Institutional Review Board (FWA00000572, expires May 31, 2014)
PROTOCOL TITLE: Dissertation Research Project
IRB NUMBER: 10-180

Effective January 29, 2013, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Continuing Review request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/ responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:
Approved As: Expedited, under 45 CFR 46.110 category(ies) 5,6,7
Protocol Approval Date: February 25, 2013
Protocol Expiration Date: February 24, 2014
Continuing Review Due Date*: February 10, 2014

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:
Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Informed Consent for Participants in Research Projects Involving Human Subjects

Title of Project
Agile Usability REU 2010 Team Study

Investigator(s)
Dr. Scott McCrickard (Associate Professor, Virginia Tech) – Principal Investigator (PI)
Jeremy Barkdale (PhD Student, Virginia Tech) – Co-PI

I. Purpose of this Research/Project

The purpose of this project is to collect data from participants about the team’s interaction. It is anticipated that the data will be analyzed and published in a conference proceeding or journal. The anticipated findings are feedback on various team structures.

II. Procedures
The procedures involved in this project are for the participants to receive training on the process components and use the processes to develop a software application. During the project, you are expected to document your experiences, time, and related tasks.

III. Risks
No risks, outside of those associated with normal daily work activity, are anticipated. You may withdraw from the study at any point.

IV. Benefits
The intangible benefit for participating in this study is the continued progress towards providing better agile usability methods for the software and usability engineering communities as well as your experience in working on an agile usability project.

No promise or guarantee of benefits has been made to encourage you to participate.

V. Extent of Anonymity and Confidentiality
Your personal information will be kept confidential. This will be accomplished by minimizing the amount of identifying information collected.

It is possible that the Institutional Review Board (IRB) may view this study’s collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research.

VI. Compensation

[NOTE: Subjects must be given a complete copy (or duplicate original) of the signed informed Consent.]

Virginia Tech Institutional Review Board: Project No. 10-180
Approved February 20, 2013 to February 24, 2014
No compensation will be provided for this project. Any expenses accrued will be your responsibility and not that of the research project, research team or Virginia Tech.

VII. Freedom to Withdraw

You are free to withdraw from the study at any time without penalty. You are free not to answer any questions or respond to experimental situations without penalty.

VIII. Subject’s Responsibilities

You voluntarily agree to participate in this study and have the following responsibilities:

1. To actively participate in the development project.
2. To follow the processes as prescribed, unless otherwise stated.
3. To provide the required feedback throughout the development project.

IX. Subject’s Permission

I have read the Consent Form and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent:

____________________________________  Date ______________
Subject signature

Should you have any pertinent questions about this research or its conduct, and research subjects’ rights, and whom to contact in the event of a research-related injury to the subject, you may contact:

Jeremy Barksdale
Investigator
336.392.6248/barksdale@vt.edu
Telephone/e-mail

Dr. Scott McCrickard
Faculty Advisor
540.231.6698/mccricks@cs.vt.edu
Telephone/e-mail

Dr. Barbara Ryder
Departmental Reviewer/Dept. Head
540.231.8452/ryder@vt.edu
Telephone/e-mail

Dr. David Moore
Chair, Virginia Tech Institutional Review Board for the Protection of Human Subjects
Office of Research Compliance
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, VA 24060
540-231-4991/moored@vt.edu
Telephone/e-mail

[NOTE: Subjects must be given a complete copy (or duplicate original) of the signed Informed Consent.]

Virginia Tech Institutional Review Board: Project No. 10-180
Approved February 25, 2013 to February 24, 2014