2016

AN ANALYSIS OF STAKEHOLDERS COMMUNICATION IN COLLABORATIVE SOFTWARE DEVELOPMENT PROJECTS

Wei Zhang
Michigan Technological University, wzhang5@mtu.edu

AN ANALYSIS OF STAKEHOLDERS COMMUNICATION IN COLLABORATIVE SOFTWARE DEVELOPMENT PROJECTS

By

Wei Zhang

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Applied Cognitive Science and Human Factors

MICHIGAN TECHNOLOGICAL UNIVERSITY

2016

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Applied Cognitive Science and Human Factors.

Department of Cognitive and Learning Sciences

Dissertation Advisor: *Dr. Robert L. Pastel*

Committee Member: *Dr. Shane T. Mueller*

Committee Member: *Dr. Kelly S. Steelman*

Committee Member: *Dr. Alex Mayer*

Department Chair: *Dr. Susie Amato-Henderson*

Contents

List of Figures

List of Tables

Preface

Part of the contents in Chapter 1, 2, 3 and 4 have been published in the paper entitled "Communication is More than Verbal: The Role of Clients' Documents in Requirement Solicitation" (Zhang & Pastel, 2014). I am the author of the paper. I did the literature review, data collection and analysis, and writing the paper. The coauthor, Robert Pastel, guided me in the research, reviewed my manuscript and gave comments.

The second section of Chapter 5 contained the content from the published paper entitled "Team Building in Multidisciplinary Client-sponsored Project Courses" (Pastel et al., 2015) and "Interdisciplinary Team Collaboration between Software Engineers and Technical Communicators" (Zhang & Pastel, 2015), of which I am the author. For the first paper, "Team Building in Multidisciplinary Client-sponsored Project Courses" (Pastel et al., 2015), I did the data collection and analysis, most of the writing was done by Robert Pastel. Another author, Alex Mayer, is the principle investigator of the project. The other author, Marika Seigo, is the collaborator of the course and contributed to part of the writing regarding the course collaboration. For the second paper, "Interdisciplinary Team Collaboration between Software Engineers and Technical Communicators" (Zhang & Pastel, 2015), I did the literature review, data collection and analysis, and writing the paper. Robert Pastel provided

me guidelines and good comments.

Acknowledgments

During the course of my PhD study, a number of people were very important in helping me with my degree. First, I would like to extend my most thanks to my supervisor, Dr. Robert L. Pastel. I started working with him in the second year of my PhD study. I still remember the first day when I met him, he introduced me to the course and the project. I appreciate that he let me be part of the project and conduct research in the course. Dr. Pastel's experience, views and insights into this research have guided me through the course of my PhD study. Without his guidance, I could not imagine how I can finish the degree. Apart from the guidance in academics, Dr. Pastel is also a kind and considerate mentor. Every time I encountered challenges in studies, work or life, he always gives me generous support and help.

Second, I want to thank Dr. Shane T. Mueller. He brought the R language into our program in 2012. I felt very lucky to be one of the students in his first year statistics class at Michigan Technological University. We not only learned statistical techniques, but also R language. After Shane's course, I fell in love with R and continued to explore this language. Other than that, my gratitude is also for the two projects that I have worked with Dr. Mueller: one is to develop an objective visual metrics for the Internet Corporation for Assigned Names and Numbers (ICANN), the other is to develop a cultural model for the Air Force Research Laboratory, not only because of the financial assistant I got to continue the PhD studies, but also because these projects have provided me in cognitive modeling apart from my main research and made me more skilled.

I would also thank my husband, Hao, for his love and support throughout my PhD course. Hao is working in high energy particle astrophysics. It is interesting that though we are working in different disciplines, some of our research methodologies are similar. I feel very happy to have him in my life because we share many values in academics and career goals. I also would thank our son, Carl. I feel sorry for spending a lot of time on work after you were born last year, but you are such a great baby who is alway making efforts to adapt to your busy mom. From the family side, I also want to thank my parents for teaching me to be a hardworking, earnest and independent person and thank my parents-in-law for their generous financial support to my PhD studies.

Finally I would like to thank my other committee members, Dr. Kelly S. Steelman and Dr. Alex Mayer, for giving me help and great suggestions on my thesis. Without them, this thesis would not be as good. More thanks go to the program of Applied Cognitive Science and Human Factors, the Department of Cognitive and Learning Sciences and all of the families in my department. Thanks Ket for being my best office mate.

Abstract

Software development is a multidisciplinary collaboration involving many stakeholders. However, existing software development processes exhibit many issues related to that collaboration. Because prior research on stakeholder analysis and teamwork revealed the importance of communication, this study analyzed stakeholder communication with reference to team activities as a social and cognitive process. The study's goal was to understand the collaboration process during software development and to delineate factors that influence this process. We focused on communication between the software developers and their clients during the requirements gathering phase, the team process, and the inter-team and interdisciplinary collaboration, in particular between software engineers and technical communicators. First, we conducted observations to help uncover the causes of variances in collaboration performance. Then we modified aspects of the collaboration process and compared team performance. We also performed an experimental study to further test the supporting effect of clients' documents on requirement gathering. Finally, teams' working structures and their impact on team performance were investigated using social network analysis. Among our findings was that clients are critical to the success of software development. Providing teams with documents that support requirement gathering facilitates team efficiency, but there is a trade-off in that team members may generate fewer creative ideas. Another finding was that software teams should ensure

that members from all disciplines actively participate in projects. Finally, although teams need leadership, effective leadership is not a strong team member performing all coordination and tasks. A moderately centralized team structure is preferred.

Chapter 1

Introduction

Software development is a creative collaborating process of multiple stakeholders from many disciplines. [1] The development process generally begins with the planning and preparation of a software product by the product owners. Development teams then gather requirements from clients, design the interface, and implement the design. The process is iterative until the product is ready for usability tests. The Agile development process emphasizes more on iteration and practice. Through iterations, teams get more experienced in self-directing the development process. Project managers do not lead the team, rather they facilitate the development process (Cockburn, 2006). Very often teams have the wisdom to get the best solutions to problems. However, sometimes they can easily be confused, because a team is a very complex system

[1] Part of the contents presented in this chapter have been published in the journal of Human Factor and Ergonomics (Zhang & Pastel, 2014).

impacted by many human factors.

1.1 Problems in Software Development

This innovative process usually collects broad ideas and creative solutions that need expertise from more than one discipline. Challenges that occur during software product design and development usually cannot be solved from the perspective of only one knowledge domain.

The collaborating stakeholders are groups or individuals who can affect or are affected by a project (Alexander, 2005; Alexander & Robertson, 2004; R. B. Freeman & McVea, 2001; R. B. Freeman & Medoff, 1984). A typical modern software development team includes product owner, graphic designers, user experience designers, technical communicators and engineers, along with other stakeholders, such as clients and project managers.

The collaboration of these stakeholders from various disciplines has helped to produce many successful software applications. However, the interdisciplinary collaboration has also created challenges. Many of the challenges are due to the communication barrier across the disciplines. According to Mathis (2009), one software project manager, stakeholders often complain about the difficulty of building a shared mental model with software engineers, not only because of the interdisciplinary communication barrier, but also because software engineers are often unwilling to invest time and effort into communication as they consider programming to be more important. Software project managers in industry have considered employing designers with programming backgrounds to enhance communication in development teams. However, this creates other issues caused by constrains of programming - assuming a designer often considers the programming difficulty in implement a feature of an app when prototyping the interface, his design ideas might be confined. The programming technology is used to implement the design, rather than being a prerequisite of the design. In addition, the agile development practices involve more direct, informal and constant communication between team members than traditional software development process, which has brought software engineers challenges, because many software engineers may be more comfortable in coding than talking (Conboy et al., 2011; Leon, 2004). These issues and challenges have led us to seek a way to make software developers and other stakeholders work as a team and to facilitate the team's collaboration.

1.2 Stakeholders Communication Analysis

Stakeholder modeling and analysis have been used in many other fields, such as public policy (Roberts & Bradley, 1991), natural resource management (Grimble & Chan, 1995), information system development (Pouloudi, 1999), and science projects

(Hein et al., 2011). Researchers in psychology, management, and computer science have studied software development process in terms of stakeholder collaboration for decades. They have investigated the communication, collaboration, and coordination processes in software development to identify factors, fidelities, and architectures that would help improve the process and create a more productive development mode. For example, Yen et al. (2001) has developed a multi-agent architecture, called CAST (Collaborative Agents For Simulating Teamwork), to support proactive information exchange in a dynamic environment. The application indicates that the architecture has enhanced the effectiveness of teamwork among agents and supported flexibility and dynamics in teamwork and role selection at run time.

Communication is usually considered to be a component of collaboration. Generally, collaboration needs to result in promising and creative outcomes such as new products, while communication is simply the sharing of information that already existed (Jackson, 2010). Over time communication within teams can become collaboration, and tools that facilitate communication can effectively support collaboration. In the software development process, communication is considered as a central mechanism of information processing at the team level (Cooke et al., 2004; Salas et al., 2008). The process needs team members to continuously share and integrate knowledge throughout all development phases. In a regular basis, a team usually has a stand up meeting everyday. All members need to present in front of the whole team what he has accomplished yesterday, what the plan is for today and if there are any challenges. Apart from the stand up meetings, there are sprint planning and reviewing meetings, and more constant casual communication in the workday as well. All the communications consist of the development process and could determine the deliveries of this process. Saeki (1995) has investigated communication between stakeholders throughout the entire software development process and determined that the effectiveness of the communication medium is dependent on the specific development phase. The product planning and design thinking could start with communication via verbal conversations, Emails or telephones, by which information such as text, voice or figures can be exchanged. In a further developing stage, a larger artifact is needed. They recommend a workspace that every team member can access, update, sit together and perform task cooperatively.

1.3 Goal of the Study

The purpose of the study is to verify the interdisciplinary collaboration issues and seek ways to solve these issues, and to build a comprehensive architecture for understanding the collaboration and team processes during software development, thus to support and improve the development of software (along with other innovative, interface design-related products).

In this research, we will study the communication between stakeholders in different phases of software development, including requirement gathering, design, and implementation. Regarding interdisciplinary collaboration during the development process, we will investigate team shared mental model across disciplines by studying team members' motivation, satisfaction, retrospection of team process and peer evaluation. Team performance will be measured in terms of its two dimensions, team efficiency and effectiveness (Salas et al., 2008). Other influences from individual variances such as personalities and skills will be investigated to ensure the construct validity. We observed and studied the real development teams and understand their complex communication patterns, rather than placing sample groups in an artificial and simplified experimental environment and controlling treatments. However, this observation of the natural data of human characteristics and performance may bring other issues of the real-world problems. Lots of factors are beyond our control to be delineated.

1.4 Thesis Layout

The rest of the thesis proceeds as follows: Chapter 2 reviews the literatures, including basic theories in team analysis and some methods that people have already used in studying teams and stakeholders' collaboration. Chapter 3 introduces the context of the study, research questions, and the techniques used to explore the answers for the questions. Chapter 4 discussed our investigations on the collaboration between software development teams and their clients, including an ethnographic study conducted in 2013. Besides, a follow-up experimental study which was conducted in 2015 was also described in Chapter 4. Chapter 5 presents the intra-team analysis in software development, including social network analysis on team structures and dynamic team process investigation. Chapter 6 concludes of this thesis.

Chapter 2

Related Work

Many studies have been conducted in analyzing team processes, team mental models and performance. This chapter reviews the related literatures on team analysis and stakeholder collaborations in software development and other fields as well. [1]

2.1 Team Studies

Dyer (1984) defined teams as "social entities" consisting of members with "high task interdependency" and shared common goals (Salas et al., 2008). Team members integrate and share information, coordinate the task, and collaborate with task demands

[1]Part of the contents presented in this chapter have been published in the journal of Human Factor and Ergonomics (Zhang & Pastel, 2014, 2015).

shifting in order to complete their mission. A number of studies on teamwork and collaboration have focused on investigating team mental models. The approaches that people use to study teams range from capturing individual team members' minds to examining dynamic interactions between team members (Klimoski & Mohammed, 1994; Mohammed et al., 2010; WIldman et al., 2014). As a common measurement of team outcomes, team-related performance has been measured through either its internal team dynamics or individual team members contribution (Salas et al., 2008).

2.1.1 Capturing Team Mental Model

A team shared mental model is described as the overlapping mental representation of a team, explaining how a team deals with difficulties and task-constrained environments (Cannon-Bowers et al., 1993; Mohammed et al., 2010; Uitdewilligen et al., 2013; WIldman et al., 2014). Team shared mental models are often computed in terms of similarity or accuracy. The individual mental models of team members can be collected and compared with each other. We can also compare team members' mental model with an expert's model so as to determine the accuracy of the team mental model. A team mental model can also be represented in the team's dynamic interaction during a collaboration process. A new stream of literature on team cognition is to consider team as a dynamic process, and the team mental model is conceptualized as "communication exchanges" that occur between team members

(WIldman et al., 2014).

Below introduces a brief history and major approaches of studies on team mental models. To date, three approaches to capture team mental models have been classified: naturalistic approach (Avnet, 2009; Klein, 1998), collective approach (Cooke et al., 2000), and holistic approach (Klimoski & Mohammed, 1994; Mohammed et al., 2010; WIldman et al., 2014). The naturalistic approach is based on naturalistic decision-making (NDM), which focuses on how experts make decisions in the complex scenarios in the real world. This approach considers team cognition as an individual mind (Avnet, 2009). The collective approach views shared knowledge as a "collection of the knowledge" of individual team members (Cooke et al., 2000). This approach is often found in social psychology and organizational behavior studies. However,Klimoski & Mohammed (1994) argued that team shared knowledge is more than the collection of individual team members' knowledge and results from the interaction process among team members. This is known as the holistic approach, which deals with the dynamic team interaction process including communication, coordination, and situation awareness. This approach is often used in human factors research and requires new methods for data collection, for example, interviewing the team as a whole to learning what the team knows. Until now psychologists have emphasized that a team shared mental model does not only indicate the stable knowledge representation of objects or situations, but also considers how information changes in a complex context (Mohammed et al., 2010; WIldman et al., 2014). The naturalistic and collective approaches both underestimated the importance of the team interaction process in determining what the team knows (Cooke et al., 2000). An aircraft crash that occurred due to communication failure provides a strong example of the importance of the interaction process. In 1990, flight Avianca 52 ran out of fuel while approaching John F. Kennedy International Airport in New York. The crew asked for an emergency landing using the word "priority", but due to the language differences between English and Spanish, the crew failed to declare the emergency and the plane crashed. In Spanish, the word "priority" can be interpreted as an emergency but that is not the case in English. In this situation, both the crew and the staff on the ground may have had the skills to identify the emergency and deal with it, but they failed to clearly share what was on their minds in terms of the situation during the interaction process. The miscommunication was fatal. 73 out of 158 people were killed (*Avianca Flight 52*, 1990). This was a complex, task-constrained scenario requiring higher cognitive processes. Although not causing death, similar problems in the interaction process in a design-related team activity can result in project failure.

2.1.2 Team as a Social Process

Focusing on the interactive process, some researchers on design-related teamwork regard teamwork as more of a social and cognitive process than as only a technical process (Cross & Cross, 1995; Olson et al., 1992). In 1992, Olson et al. (1992) studied ten design meetings in two organizations and found that the design meetings are not all about designs - only 40% of the meeting time was spent in direct discussions of designs. The remaining time was spent on walkthroughs and summaries (30%) and coordination (20%). In general, at least a third of the communication was for clarifying ideas while working on designs or preparing for walkthroughs and writing summaries. The meetings should be mostly used to communicate and exchange ideas on project, rather than to coordinate team members' schedules.

In addition, the social processes may influence many aspects of team activity. In Cross' observations of a three-person team design session for the Delft Protocols Workshop (Cross & Cross, 1995), he saw that some team members' approaches were ignored during project planning, a phase that needs very intensive communication for generating and gathering a variety of ideas. When a team member experiences difficulty in getting the team to proceed in a way he prefers, the team member could feel very frustrated and isolated, which could influence his contribution to the team. The roles and relations are forming during the team process, and such social process interacts significantly with the technical and the cognitive process of design.

Team conflicts can also be avoided or resolved by skillful social techniques. Galegher & Kraut (1994) examined the difficulty of using computer-mediated communication to accomplish complex collaborative work by observing MBA student groups completing collaborative writing projects under three communication conditions: computer only, computer plus phone, and face-to-face communication. The findings showed that individuals can adapt to the restricted communication channels, but lack of interpersonal interaction leads to difficulties involving ambiguous goals and diversity of ideas. Some tasks in particular, such as project planning, may require more intensive communication than independent work.

When studying team activities as a social process, social network analysis has been shown to be an excellent tool for analyzing the team mental model (Espinosa & Clark, 2014). When team members across domains do not share their knowledge, the measurement of team knowledge becomes more complex than simply averaging and providing an incomplete picture of the knowledge structure (Avnet & Weigel, 2013; Espinosa & Clark, 2014; Newman, 2003). In a cross-discipline team, team members may shared certain common knowledge, such as a common project goal. Apart from the shared knowledge, there is role-specific disciplinary knowledge that is compatible among team members. The interaction of the knowledge creates the team dynamics. Network experts (Espinosa & Clark, 2014) considered team knowledge as an inherent social construct, in which individual social actors share and exchange knowledge through communication and create a cognitive relationship to build the dynamic interaction process. Espinosa & Clark (2014) created the network model with a quantitative metric of change in the team shared mental model. Through network analysis, Avnet & Weigel (2013) demonstrated that in engineer design projects, i.e. spacecraft launch project, team shared knowledge correlates the technical design of system attributes, including system development time, system mass and technological maturity. Although they did not examine team performance directly, it provides insights of improving system design product by increasing shared knowledge and incorporating it into design process. Espinosa & Clark (2014) found that network structures explained how knowledge flows within a team and influences the coordination and team performance. Their networks showed that low performing teams have more disconnected shared task knowledge between team members (See Figure 2.1). In our study, we will use social network analysis to investigate the communication structure between stakeholders.

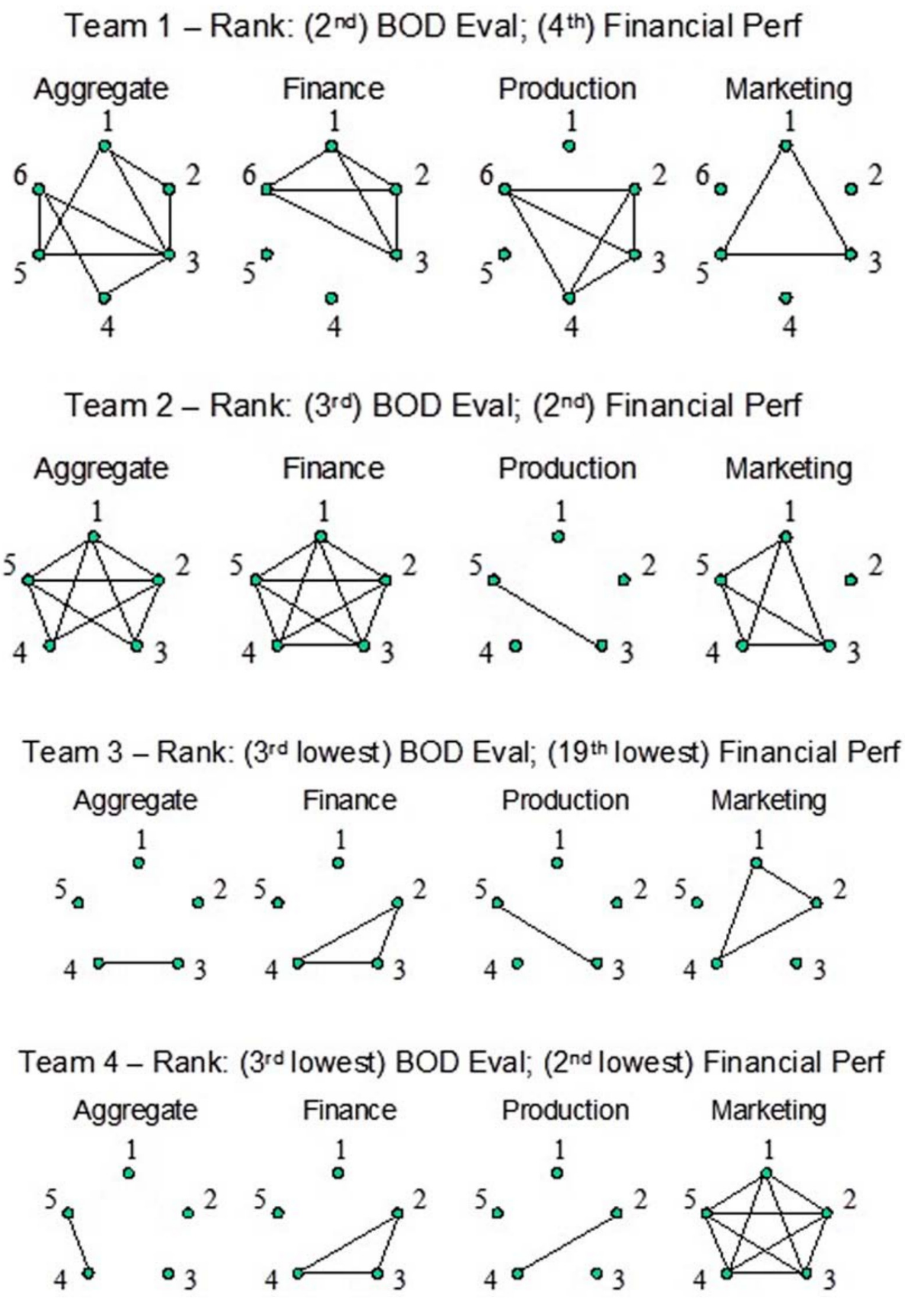

Figure 2.1 Illustration of Shared Task Knowledge for High and Low-Performance Teams (Espinosa & Clark, 2014). Team 1 and 2 are high-performance teams; team 3 and 4 are low-performance teams.

2.1.3 Measuring Team Performance

Team effectiveness and team efficiency are two essential dimensions in evaluating team performance. Team effectiveness is an appraisal of the outcomes of the activities engaged in while completing a task (Hoegl et al., 2003; Salas et al., 2008). Team efficiency relates to project budget and schedule. It is usually measured by cost and time-to-completion (Faraj & Sproull, 2000; Hoegl et al., 2003; Jones, 1996). Studies show that the administrative coordination influences team performance (Faraj & Sproull, 2000; Kraut & Streeter, 1995; Van de Ven et al., 1976), especially team efficiency (Ancona & Caldwell, 1992; Faraj & Sproull, 2000). Administrative coordination refers to the traditional mechanisms used to coordinate teamwork, such as the use of milestones to assign tasks, allocating resources, and integrating output. Faraj & Sproull (2000) proposed that *expertise coordination*, as a set of socially shared cognitive processes, is more essential than administrative coordination to the performance of *knowledge teams*, such as software development teams. Another study conducted by Hoegl et al. (2003) employs the teamwork quality construct as a comprehensive measure of the quality of team collaborations. The study concluded that teamwork quality is associated with team efficiency only in projects posing high task innovation, e.g. task novelty, complexity, and uncertainty. However, the relationship between teamwork quality and team effectiveness is not moderated by task innovation. Software development teams are highly knowledge- and expertise- dependent teams, but are usually formed temporarily depending on product requirements and developers' availability. Not many studies on software development teamwork have measured team performance in terms of both team efficiency and effectiveness (Driskell et al., 2006; Edwards et al., 2006; Lim & Klein, 2006; Mathieu et al., 2000; Moe et al., 2010). In this study, we investigate both.

Studies have provided much evidence supporting the fact that the sharing team mental model has a positive influence on team performance, in particular on team effectiveness (Edwards et al., 2006; Espinosa & Clark, 2014; Jobidon et al., 2012; Lim & Klein, 2006; Mathieu et al., 2000). Shared leadership and language features in communication were two factors that have been indicated as impacting team performance (Gonzales et al., 2010; McIntyre & Foti, 2013). Driskell et al. (2006) also looked at the relationship between team member personality and team effectiveness. They provided a very comprehensive prediction to link certain personalities traits, such as dominance, to different task requirements. A dominating trait may be problematic for a team member when cooperating with other members, but useful when the team member becomes a team leader. Recently, Moe et al. (2010) looked at the interrelations within the agile software development teams and found that proper team orientation, coordination, and work division are essential for successful software development. Though the agile teams are self-directed, they need orientations by both the team members and manager, e.g. the priority to team goals. The manager gives his team direction to work toward and provides as many resources as possible to facilitate the development process. Along with these factors, we expected that there would be other influences, such as affective status, motivation, professional skills, and task complexity. Researches have indicated that positive affective states can lead cognitive process to be broad whereas negative affective states narrow cognitive process (Easterbrook, 1959; Harmon-Jones et al., 2012). Will this difference impact the cognitive process during software development. In education psychology, strong motivations can increase learners' effort and persistence in activities (*Essentials of Educational Psychology*, 2009). Will team members who are more motivated contribute more? The task complexity and team members' programming skills could be factors that influence the completion time and product quality. Previously, no study has comprehensively explored factors that potentially influence team mental models during the software development process.

2.2 Interdisciplinary Collaboration in Software Development

Interdisciplinary collaboration became popular in business and academia decades ago. In 1986, Curtis et al. (1986) called for interdisciplinary collaboration in the software industry, in particular the collaboration between software engineers and human factors experts. Human factors experts work to develop models and theories of fundamental characteristics of human behaviors that specifically describe interaction with the software, while engineers implement these models into actual designs. In healthcare systems, interdisciplinary practice is also very important because it often involves nurses and physicians with different specialties. Orchard et al. (2009) proposed a patient-centered interdisciplinary collaborative professional practice model to accommodate shared decision-making during team process. In a longitudinal study, Cashman et al. (2004) reported the supports and barriers brought by teamwork can affect the development of an integrated health care system. In addition, scientists have conducted research into collaborating with colleagues from other disciplines. Interesting new measures and models have been developed by interdisciplinary collaborations and have advanced science (Borgman & Furner, 2002; Qin et al., 1997). Because software development is an innovative process collecting broad ideas and creative solutions which requires expertise from various disciplines in terms of the product design, we believe that the interdisciplinary collaborations between stakeholders and across functional teams are as important and indispensable as in other areas. Phuwanartnurak (2009) has suggested wikis as tools to help with information sharing and scaffold interdisciplinary team collaboration. In addition to these tools, we want to explore more how the communication and coordination within software development teams can be improved.

Chapter 3

Research Method

This section introduces the study's context, the research questions and the standard methods used to explore the research questions. [1]

3.1 Study Context

Salas et al. (2008) advocated that studies on teamwork performance should focus on more teams "in the wild", i.e. in a fully situated context. The context of our study is a citizen science project funded by the National Science Foundation (Mayer et al., 2013). One project goal is to develop Android applications (apps) or websites that

[1]Part of the contents presented in this chapter have been published in the journal of Human Factor and Ergonomics (Zhang & Pastel, 2014, 2015).

anyone can use to collect data in the field for scientific use. Students from collaborating courses at Michigan Technological University work as groups to develop the citizen science apps or websites. The current study is based on the observation of their development processes from 2013 to 2015.

Four years ago, considering that computer science students have few opportunities to work with people from other disciplines and with real clients, Dr. Robert Pastel integrated the citizen science project with a collaboration of three courses: two Computer Science (CS) courses in Human-Computer Interaction (undergraduate and graduate courses) and a Scientific and Technical Communication undergraduate course in the department of Humanities (HU). Students in the two undergraduate courses worked in teams developing the apps, while students enrolled in the graduate course worked individually to evaluate the designs and test the usability of the finished apps. The development teams worked with professors and scientists from around the U.S., who served as both clients and domain experts for each app. This collaboration involved stakeholders with many disciplinary backgrounds and required students to have both task-relevant professional skills and skillful collaborating approaches when working with clients and teammates from other disciplines.

The teams practiced software development process in phases, which includes user-centered rapid prototyping. Assignments in the course fixed the approximate timing of the development phases. The instructors for the courses were program managers and determined the approximate development schedule. However, there are some concerns of the study context. One aspect of development in the context of a course that differs from development in industry is that many of the developers are inexperienced with the development technology, although it is not unusual for development teams in industry to have inexperienced members. Another aspect that differs between software development in a course and industry is that students may have little experience communicating with clients; however, during course preparation, the instructor met with each scientist to ensure that a well-established app idea was developed and to initiate requirement gathering.

3.2 Research Questions

In this context, our investigation is focused on the collaboration of these diverse development teams and the client participation. Below are the two research questions that we strived to answer:

Question 1: What are the main collaborative factors that determine or influence software development process?

Question 2: How can we improve the collaboration in software development regarding these factors?

3.3 General Methods

A number of study techniques were used to answer the research questions. First, we made observations using ethnographic techniques to investigate the communication and development processes. Second, we distributed several surveys probing communication among team members and between teams and their scientists at the end of each semester. In addition to the surveys at the end of each semester, social network analysis was used to study the teams' working structures. In order to learn how these team working structures influenced their performance, we use a team process survey to investigate these individual perceptions of team members relationship as a measure of performance variance.

For the 2014 and 2015 semesters, we made modifications to the course based on the insights from previous years. The modifications include changing the class locations, combining the two classes by giving lectures by one of the instructors, distributing agreement between scientist and teams etc. The comparison was made from the responses to the semester surveys each year. Moreover, scientists provided the development teams a document, which is usually a form that the scientists use to collect actual data in their domains, including time, the features of the current environment such as number of trees etc. In order to test the effects of these documents, we implemented an experiment in a laboratory in the Fall of 2015. In the experiment, student participants work in teams to design the first prototype of an app interface. They were organized into two types of teams, composed of only CS-related disciplines, and mixed CS and other disciplines. Two experiment scenarios were given: groups in the controlled scenario did not receive supporting document, and groups in the experimental scenario did not receive it.

3.3.1 Materials

In the ethnographic study, we collected data through surveys, email records, semi-structured interviews, and documents produced by the teams; In the surveys, we focused on communication because we expected that communication would be a key factor that would cause performance variances. The surveys also collected team members' motivation, satisfaction and ratings on other members' performance. (Appendix B is the survey for CS students in 2014, the survey for HU students was similar).

We used cognitive social structures (CSS) survey to collect data for social network analysis. The survey is to collect team members' individual perceptions of the relationships among each other, such as the perceptions of which team member will go to the other for help (see Appendix H). To measure the performance variance, the team process survey, which is similar to the semester survey mentioned above, was given to students three times during the 2015 semester. One example is shown in Appendix I.

3.3.2 Participants

In general, there are 99 students (76 male and 23 female) who have registered the course from 2013 to 2015. These students worked in 17 teams, including five development teams in 2013, six in 2014, and six in 2015. Each team comprised five to eight team members, including four to six CS team members and one to two HU team members. The teams in 2013 were Lichen, Stream, ROV, Beach, and Tracking; the teams in 2014 were Watershed, Mega Crystal, Tree Walkers, Thunder Bay, Google Fox, and Water Level Wizards and the teams in 2015 were Fisheye, CoCo, Field, Bear, Deer and Ice.

The development teams were organized during the first week of the semester. Although teams were primarily self-organized, the instructor confirmed that each team had at least one member with Android experience. Each team had one or two scientists who were both clients and domain experts. In 2013, the Lichen team had two scientists. One was remote, meaning that it was not possible to have face-to-face communication; the other scientist was local. The Beach team had one remote scientist. The other three teams all had one local scientist. In 2014, all teams had remote scientist, except that Watershed team's scientist was local. In 2015, all the 6 teams had remote scientists. The teams chose which app to develop. All the citizen science applications were environment related and were intended to enable citizens to gather information from the local community environment for scientific purposes. For example, the Lichen team developed an app for users to observe and collect data on the distribution of lichen species, which can be used as an air quality indicator. The Stream team developed an app for users to record the condition of stream health and erosions at road crossings.

In total, twenty-five groups of students (79 students) participated in the laboratory experiments to test the supporting document effect, with three students in each group: ten control mixed-discipline groups without document and ten experimental mixed-discipline groups provided with document, and another five single-discipline groups to test the diversity effect. Most participants for the experiment were recruited from the senior classes of the Department of Computer Science. Apart from that, the participants other than computer science students were from the senior classes of the Department of Humanities and the Sona systems subject pool at Michigan Technological University. We required that students who have taken the HCI course cannot participate the study in order to eliminate the bias that they may have by having been trained in the course and know more about the design process.

Chapter 4

Collaboration between Development Team and Client

A critical phase of software development is requirement gathering, which involves communication between users, clients, software developers, and managers. [1] All the citizen science apps in our project were environment related and were intended to enable citizens to gather information from the local community environment for scientific purposes. Anyone can use the apps to submit data in a scientific field. Finally the data collected will be used by the scientists. Therefore, the scientists are not only domain experts, but clients as well. We decided to first study the communication between the development teams and clients during the requirement-gathering

[1]Part of the contents presented in this chapter have been published in the journal of Human Factor and Ergonomics Zhang & Pastel (2014).

phase. We suspected that their communication about requirements is important to the citizen science application development. If developers misinterpret requirements, the scientists will ultimately not use the application and the application will lose its value.

Related work in requirement gathering shows that errors in requirements can cause considerable delays and possibly result in project failure (Sommerville & Sawyer, 1997). About one-third of software projects are completed without gathering sufficient requirements from clients (Chatzoglou & Macaulay, 1996). In global software projects, the lack of informal or face-to-face communication can have a negative impact on requirement gathering (Damian, 2007). It could be more difficult to schedule a time with a remote clients to collect requirements. Relying too heavily on indirect links, such as customer surrogates, to gather requirements can also lead to project failure because information could be easily lost when the message is passing between more people (Keil & Carmel, 1995). Consequently, the communication between developers and clients is critical during the requirement gathering phase of software development and can determine the success of the entire project. This chapter describes an ethnographic study conducted in 2013 and 2014, and an experimental study conducted in 2015, focusing on the communication and collaboration between the software development teams and their clients.

4.1 Study I, Observation on the Collaboration Between Software Development Teams and Clients, 2013-2014

In Study I, we observed the communication between developers and scientists during the development of five citizen science apps in 2013, with a comparison of the development process of the six apps in 2014. Our goal was to discover possible deficiencies in the communication or development process and to delineate critical factors that led to success.

4.1.1 Hypothesis

We suspected that communication could be a direct measure of collaboration, that the communication between software developers and clients is a critical factor to the success or failure of a project, and that more communication between the two stakeholders is better. Our hypotheses for this study were:

Hypothesis 1: The communication between software developers and clients can predict better requirement gathering.

Hypothesis 2: The quantity and quality of communication between software developers and clients can positively impact team performance during the development process.

4.1.2 Methods

We made many observations using ethnographic techniques to investigate the communication between teams and their scientists, and development processes. We collected data through surveys, email records, semi-structured interviews, and documents produced by the teams. We distributed surveys to investigate communication between scientists and development teams across two years in order to compare the rating results and to evaluate the modifications on the course.

4.1.2.1 Participants

The observations were made on the development teams in the class in 2013 and 2014, including all the students enrolled in the class. There were 5 teams in 2013, including 19 CS students and 8 HU students, and 6 development teams in 2014, including 27 CS students and 10 HU students. All students responded the surveys in 2013 and 2014. The email records and meeting notes were collected from all the development teams. We received consent from both the scientists and developers before collecting the data.

4.1.2.2 Materials

In 2013, we distributed a survey at the end of the semester to both CS and HU students (Appendix A is the survey for CS students, the survey for HU students was similar). In the survey, we asked students to estimate the communication frequency with scientists and their teammates. For example, we asked the question "How frequently did you use EMAIL to communicate with your scientist". In 2014, we also distributed a survey to students from both disciplines at the end of the semester (Appendix B is the survey for CS students, the survey for HU students was similar). In addition to communication questions, we added questions about motivation, confidence, conflict, and self-ratings on team performance. For example, "Rate the average CS team members' performance", "how you enjoyed the development process working with your teammates". Developers also kept good documentation on their development process on the group websites. These included dated meeting notes and design documents.

4.1.2.3 Procedure

We closely observed the development team process both on and after the class in 2013 and 2014. The emails and meeting notes were collected in order to better investigate the communication process because the teams were distributed. At the end of each semester, we distributed the paper-based survey introduced in the above section. Other than these, we conducted semi-structured interviews with at least one team member and with the whole team. During the interview, we asked participants to recall the communication process with their scientists, such as how many meetings did you have with your scientist. The interviews were recorded.

4.1.2.4 Data Analysis

We estimated the quantity of communication in all media from the survey. In the survey, we asked questions "how frequently do you meet with your scientists?" and they could respond "daily", "once a week", "twice a week", "twice a month" and "once a month". We counted emails and deduced the number of meetings according to emails, meeting notes, and interviews. We also studied documents that developers received from scientists and qualitatively determined their contribution and effectiveness to the design of the app. We focused on the Lichen and Stream teams because they had a large disparity in the quantity of communication with their scientists and performance difference in the usability tests and software correctness. We determined the development progress by checking the dates and phases of the apps' development from email records, meeting notes, and interviews. To analyze the communication records, we used both content and pattern analysis. We read the content of the emails to look for information about teams' development timeline. We also use social network analysis to explore the patterns of team structures. The emails were categorized in terms of subjects, and the information flow structure among the emails was captured through social network analysis (L. C. Freeman, 1979). The course was modified in 2014 according to the insights from the observations and the survey in 2013. The rating results were compared using t-test to evaluate the effect of the modifications.

4.1.3 Results

We investigated the communication media in the survey by asking questions such as, "How frequently did you use the PHONE to communicate with the scientist?" The counts were estimated by transforming the responses. For example, if a team member respond to the frequency of meetings as once a month. Because there are about four months in a semester, we estimated the number of meetings to four according to this team members. We transformed the responses into estimated counts in order to learn the frequency of their communications in each mode. In 2013, the study started in the middle of the semester. Therefore I could not have an observation of the team communication across the whole semester. However, we could ask team members to estimate the communication frequency in the survey at the end of the semester. Their estimations should not be used quantitatively, but can be qualitatively compared. This helps us to better understand the quantity of communication in each team in general. As Figure 4.1 shows, email was the most commonly used communication medium. The Lichen team had much more communication than the other teams.

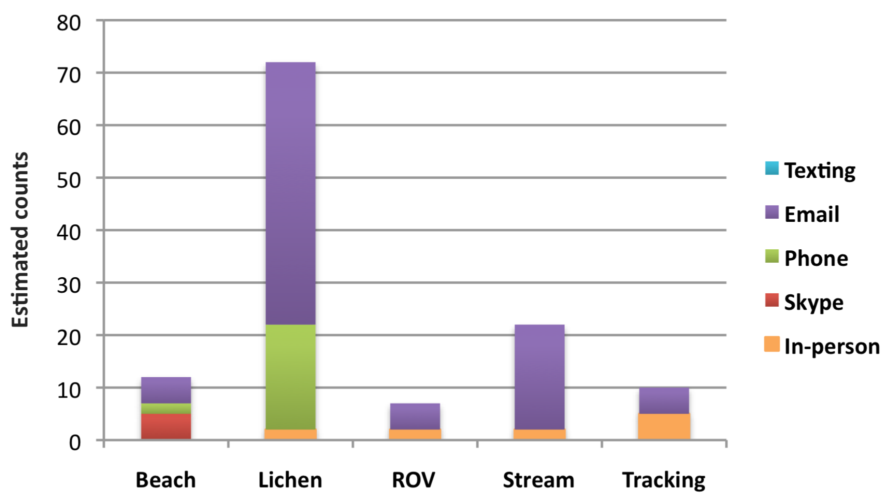

Figure 4.1 Communication Medium Distribution from Survey 2013

In order to test the estimation of the communication frequencies above, we studied all email records between developers and scientists for the five teams during the 93 days of app development. The total number of emails was 146. Table 4.1 summarizes the number of emails according to subject matter for each team. The Lichen team sent 99 emails in total, while the other teams sent no more than 20 emails. This number of emails tests what developers have estimated in the survey as Figure 4.1 shows,

Table 4.1. Summary of Email Quantities by Category

Email Categories	Beach	Tracking	ROV	Stream	Lichen	Total
Requirement Gathering	2(12.5%)	5(38%)	0	5(55%)	56(56%)	68(%)
App and Code Design	2(12.5%)	0	0	1(12%)	4(4%)	7(4%)
Arranging Meeting	12(75%)	8(62%)	9(82%)	3(33%)	42(42%)	74(47%)
Task Negotiation	0	0	2(8%)	0	2(2%)	4(2%)
Total	16(100%)	13(100%)	11(100%)	8(100%)	99(100%)	146(100%)

that the Lichen team sent more emails than the other teams. In general, most of the emails were about arranging meetings (47%) and requirement gathering (43%).

We used social network analysis to determine the structure of the email communications (L. C. Freeman, 1979). A social network is illustrated by a graph representing the communication flow. We considered scientists and developers as distinct social actors in the network, who are connected by emails. Figure 4.2 shows the email network between only the scientist and team members for the Lichen team. Other teams had similar structures. One developer sent all the emails to the scientist, and the scientist responded by emailing the entire team. We learned from retrospective interviews that the teams deliberately assigned one of their team members to coordinate communication with the scientists and relay the communication to the rest of the team.

Requirements can be communicated to the developers by documents, such as requirement lists or example forms. Table 4.2 compares the documents that the Lichen and Stream teams received from their scientists. We categorized the effectiveness of the scientists' documents at communicating the protocol and specifying the user

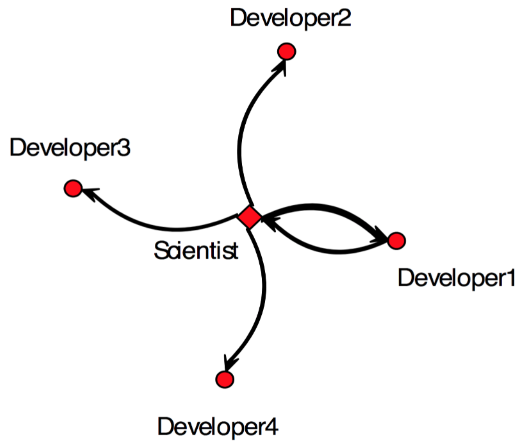

Figure 4.2 Email Network of Lichen Team 2013. The node indicates actors including the team members and the scientist of the team. The line indicates there are emails that have been sent between the two actors. The arrow indicates the direction of the emails.

interface (UI) for the app. Lichen developers received an article about lichens from the scientist, which was more than one hundred pages and difficult to translate into requirements. The developers had to ask the scientist to interpret the article so that they could translate it into requirements. The Lichen team also received a spreadsheet and pictures of lichen species late in the development phase. Interpreting the documents was a topic of Lichen's fourth meeting with the scientist. The Stream team received a stream reporting form which effectively communicated the app's protocol and served as a template for the main view of the user interface. During the interview with the Stream team, the form was referenced as the "requirements sheet". None of the three documents from the Lichen scientists appropriately served to specify both the protocol and the UI. The document from the Stream scientist was very effective for specifying both the app's protocol and UI aspects.

Table 4.2. Effectiveness Purpose of Lichen and Stream Team's Scientist Documents

Team	Scientist Document	Document's Purpose	
		Protocol	UI
Lichen	Article about lichens	No	Yes
	Spreadsheet of lichens	Yes	No
	Pictures of lichens	No	Yes
Stream	Stream reporting form	Yes	Yes

Teamwork efficiency was investigated by studying the development progress. Figure 4.3 shows a progress summary of the two teams, with the days that they spent working during each phase of the development. The initiation and completion date of each phase is based on email content, interviews with team members, and meeting notes. The progress shows five specific phases: background learning, requirement gathering, app protocol development, app design, and implementation. There were 93 days for these phases before the usability testing of the apps, which was a fixed deadline. Phases were scheduled during the course via assignments, but teams could slip on deadlines or continue working on a phase after the minimum requirements for that assignment were met. Both teams used a short period to become familiar with the domain and context of the apps. Requirement gathering is the most distinctive phase between the Lichen and Stream teams. It covered most of the development process for the Lichen team. The Stream team contacted their scientist immediately upon completion of app's domain and context, asking for requirements, but the team did not receive a response for 27 days even after sending two follow-up emails. This is depicted as a delay in requirement gathering in Figure 4.3. After the scientist's response, they finished requirement gathering in a few days. Understanding and designing the app's protocol took a long time for the Lichen team, more than 30 days. Our study of the Stream team's communication did not uncover any discussion about the app's protocol. Both teams spent the rest of the development process designing the software. The Stream team began implementation while designing. We could not find any communication indicating that the Lichen team discussed implementation. But according to the course assignment that they had deadline for finishing the project for usability testing, we deduced that due to the course deadline, they rushed all the implementation at the last minute after requirements were gathered.

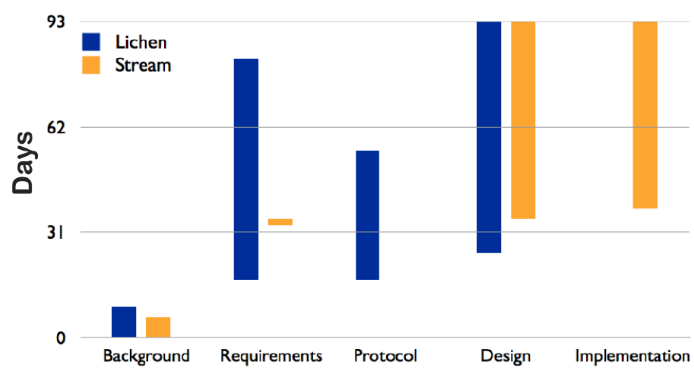

Figure 4.3 A Summary of the Lichen and Stream Team Progress.

Teamwork effectiveness is an appraisal of the outcome and quality of the delivered product by a team Salas et al. (2008). In our case, the teamwork quality could be measured in several ways. Course grades could be one of the measures, because this is a course project. However, the grades were given mainly based on the individual contribution to the project; The graduate students perform the usability tests at the end of the course. They report the evaluation of the apps including the user acceptance and software errors, which could be a measure of the quality of the apps. Therefore we determine to use the usability test reports as a measure of team effectiveness. Four of the five teams conducted usability tests. All four apps had the correct and minimal functionality according to the test reports. Focusing on the Lichen and Stream app, the Stream team did better than the Lichen team in the usability test, with less than half as many software errors and better user acceptance. In total, the Lichen app contained five errors and the Stream app had two errors. The usability tests for both the Lichen and Stream apps investigated user acceptance of the apps. The Stream team used dichotomous questions, asking if users thought the app was easy and intuitive to use. All seven users gave positive feedback by answering "yes". The Lichen team used a Likert scale, and users could strongly disagree, disagree, be undecided, agree or strongly agree. When asked if the app interface was pleasant, one respondent strongly disagreed, one disagreed, one was undecided, and three agreed. When asked if they enjoyed using the app, one user disagreed, three were undecided, and two agreed. Only one user out of six said he would use the app again. Consequently we summarize that the Lichen app's user acceptance was much lower than that of the Stream app.

4.1.4 Discussion

The Stream team had better performance than the Lichen team in both team efficiency and effectiveness. Given the data and results described above, our goal was to determine the causes of the differences between the two app's usability test results and software correctness.

As for emails, our result from the survey shows that the Lichen team had sent more than ten times as many emails as the Stream team. Although they had one remote scientist, the scientists were more responsive with emails. The Lichen scientists usually responded to emails about scheduling meetings within hours and responded to emails on requirements within several days. The responses of the Stream team's scientist were slow. One response on requirements was delayed nearly a month. Considering the number of meetings between scientists and developers, the Lichen team had twice as many as the Stream team, which suggests that the emails were effective for the Lichen team in arranging meetings and that the Lichen scientists were accessible. The Lichen team's slip in requirement gathering could indicate that email was an awkward tool for requirement gathering, but not for arranging meetings.

The Stream team was delayed in the beginning of requirement gathering because of the slow response from their scientist. But they finished gathering all the necessary requirements in a few days after receiving the information from the scientist. The Lichen team started gathering requirements earlier, but it took them two months to get all the necessary requirements. This delay impacted implementation, because the teams had only one semester and if they spent too much time on requirement gathering, they would have less time on development. We do not believe the delay was caused by the remoteness of one of the scientists because their scientists were responsive. Rather it is caused by the ineffectiveness of the document that the scientist has provided to the team.

As for the scientists' documents, the scientists for the Lichen team provided several documents to deliver requirements. The article about lichens was very long and too academic for the developers, who were not familiar with the domain. The spreadsheet informed the developers about the app protocol, but the fourth meeting late in the semester suggests that it was not easy to use. We believe that none of the three documents from the Lichen scientists appropriately served to specify either the protocol or the user interface. The scientist for the Stream team sent an actual form used by professionals to record stream conditions, which specified the exact stream traits and characteristics that needed to be implemented in the app. The document was very effective for specifying both the app's protocol and the user interface.

There might be other potential influences on team performance. In the first study, we did not measure the individual differences of stakeholders, such as the developers' and scientists' communication skills; consequently, we cannot determine how much they influenced the difference in performance of the two teams. The complexity of the apps is another possible difference that may have influenced the results. The length of the programs can be used to compare the relative complexity of the app. The Lichen app was approximately 25% longer than the Stream app according to the size of the implementation tasks. This measure of program complexity suggests that the Lichen app was slightly more complex, but we do not believe this explains all of the Lichen team's development delays. We believe that the lack of an effective document clearly specifying the app's protocol must have influenced the time spent developing the app. The scientists served as more than only clients. They are very motivated to have an actual app and have agreed with the course instructor on the responsibilities on providing the teams domain expertise and supports during the development process. Communication between development teams and clients is a critical factor in requirement gathering during software development and is more varied than simply verbal. Moreover, documentation could be another way of communication. Contrary to our initial beliefs, sending more emails does not necessarily lead to better performance. Appropriate and understandable documents delivered by the client can serve an important role in the communication of requirements and are sometimes critical for a successful implementation. Therefore, it is the quality, rather than quantity, that is important to effective communication between software developers and clients in requirement gathering. The high quality of the communication makes scientists and the teams to reach a common goal and a shared understanding of the app requirements, mainly that scientist were able to communicate what app features they want. Our findings also suggests the critical role played by clients throughout the development process.

4.1.5 Modifications and Results

As we hypothesized that the communication between software teams and clients can predict better requirement gathering. The case study of Lichen and Stream team indicates that the their communication is critical to requirement gathering phase. Ineffective communication can cause delay of gathering necessary requirements. We also hypothesized that the quantity and quality of communication can impact team outcomes. It turns out that more communication does not necessarily bring better team performance. It is not quantity but quality of communication that positively influence team outcomes. The findings of the previous study provided us with insights for ways to improve the development process, which we used to modify the course in 2014. We invested more time preparing our clients-scientists at the beginning of the process. Besides ensuring that the scientists had a concrete app idea, we searched for any documents that the scientists could share with the development teams, such as forms, reference tables, example data collections, and written descriptions of their protocols. We requested that the scientists have the documents ready to give the development team. We examined the documents' ability to communicate requirements to the development teams. The instructors reviewed the documents before the course and subjectively evaluated that wether the document communicate the requirements clearly. For example, if the document is an introduction paper on the domain knowledge, we consider it not effective document because it requires students to spend lot of time reading and figuring out the app requirements themselves; otherwise if the document is a table that the scientist uses to collect data, including all the features that the scientist wants, we consider it an effective document. If the documents were lacking, we asked the scientists to produce new documents or to revise the existing ones.

Because clients are critical throughout the development process, we explained the development process during our initial interviews with the clients. Before the development process began, we distributed a document to both the clients and development teams that detailed the process and described the roles of each stakeholder including the scientists and the teams (see Appendix C). In effect, this document was an "agreement" between the clients and the development teams. In addition, this agreement document scheduled meetings throughout the development process. Two meetings were scheduled for requirement gathering at the beginning of the development process. Two additional meetings were scheduled to the teams to present their designs and receive feedback on their designs. The first design presentation was a paper prototype and the second presentation was a higher fidelity prototype, meaning a nearly complete app. A final meeting was scheduled after the usability tests. Inevitably, clients had suggestions for improving their apps after each meeting. The agreement document explained the extent to which the app design could change at each phase of the development between the development teams and scientists. We did not require such meetings in the first year. All the meetings and communication in 2013 were depend on the teams and their scientists.

Because team efficiency refers to the time to complete tasks Salas et al. (2008). In our case we used an app completion time to measure teamwork efficiency. At the conclusion of the course, three of the six apps were completed, two apps were nearly completed, and one app was functional but not satisfactorily completed. The percentage of satisfactorily completed apps doubled from the year 2013 to 2014. In the first year, only 40% (two out of five) of the apps were satisfactorily completed, while 83% (five out of six) of the apps were completed in the second year. There was an increase in the number of teams completing their apps compared to the previous years, which we attribute in part to the changes we made on the course.

Teamwork effectiveness was also measured using the usability test results. Using the results from usability tests of the four completed apps from 2014, we compared them with the Lichen and Stream apps from 2013. Table 4.3 summarizes the results.

Table 4.3. Across-year Comparison of Usability Test Reports

Year	Teams	User Acceptance			Average Software Errors errors/person
		Positive	Neutral	Negative	
2013	Lichen	3	1	2	0.83 (5/6)
	Stream	7	0	0	0.29 (2/7)
2014	Watershed	18	0	0	0
	Mega Crystal	3	1	0	2 (8/4)
	Water Level Wizards	6	0	0	0.17 (1/6)
	Tree Walkers	8	1	0	2 (18/9)

Due to the small sample size of each test, we cannot make statistical analyses of the results, but we can qualitatively compare the number of user acceptance responses and software errors. The reports showed that the user acceptance responses improved from 2013 to 2014, but the software errors varied from different teams. We believe that there are influences from factors other than the communication of the development teams and clients on requirements, which we will study further.

Students' ratings of scientists in two years also indicated that a progress has been made. In both 2013 and 2014, we asked students to rate the communication and performance of their scientists. Table 4.4 provides the summary of the across-year comparisons. There is a significant difference between the CS students' rating of the scientists from year 2013 to year 2014 ($t(26, 1) = 4.99$, $p < .001$), but no significant difference in the HU students' communication rating with the scientists ($t(12.\ 6) = 1.86$, $p = .086$). Adjusted Hedges's g is the difference between the means scaled by the pooled standard deviation and adjusted for sample size bias Hedges & Olkin (1995). It is an unbiased standardized effect size and is a measure of the change in the effect

Table 4.4. Across-year Comparison of Communication and Performance Ratings (students⇒scientists)

Ratings	2013		2014		t value	df	p value
	Mean	sd	Mean	sd			
Communication	4.08	1.45	6.00	0.92	4.99	26.1	3.5×10^{-5}
Performance	4.13	1.70	5.85	0.95	3.90	24.1	8.6×10^{-2}

due to an intervention. In this case, the change in effect is the difference between the students' mean response on the surveys, and the intervention is the modifications made to the courses. Hedges's g (estimated value and standard deviation) for the CS students is 1.24 ± 0.3 and for the HU students is 0.91 ± 0.5. For both the CS and HU students, the modifications to the coordination and communication with the scientists had a large effect on the students' communication rating for the scientists. Therefore, we determined that students felt better working with their scientists after we modified the courses. The modifications have successfully facilitated the collaboration between the development teams and clients.

4.1.6 Summary of Study I

The goal of Study I was to verify the vital role of the communication between software developers and clients during requirement gathering, delineate critical factors in communication that led to success (i.e. the scientists' documents), and incorporate the insights into modification of the development process. We found that during the requirement-gathering phase, emails are not always the most reliable or suitable communication medium. Email frequency is not as important as we expected. Instead, the quality of communication has a larger effect on communicating requirements, and frequency of email communication may reflect poorer information sources, rather than better communication. Documents from clients are critical to facilitate the process. Explicitly, Study I demonstrated the importance of clients' documents for specifying requirements, but it also suggested implicitly the critical role of clients throughout the development process.

4.2 Study II, Experiment on Supporting Document Effect, 2015

The experiment discussed in this section was designed and implemented in order to test the effects of supporting document on the app development process. In particular, the experiment explored the documents' effects on software development completion times, the completeness of the app designs, and number of generated ideas.

4.2.1 Hypotheses

As shown in the prior study, the documents received from scientists effectively facilitated the software development process, particularly in the requirement-gathering phase. However, we were concerned that, although the supporting documents can lead to more complete solutions, too much support might inhibit the number of ideas that the teams generated. Besides, we want to test the effect of interdisciplinary collaboration in software development, therefore we have two types of teams, mixed and single discipline teams: the single discipline teams consist of team members from only CS discipline, while the mixed discipline teams consist of members from CS discipline and others such as humanity and psychology. We assumed that mixed-discipline teams can bring us more varied ideas than single-discipline teams because they have members from more than one discipline. Thus we hypothesized the following:

Hypothesis 1: The availability of supporting documents that can specify requirements will reduce the amount of time required for the software interface design.

Hypothesis 2: The availability of supporting documents that can specify requirements will lead developers to generate fewer ideas.

Hypothesis 3: The mixed-discipline teams will generate more diverse app design ideas than the single-discipline team.

4.2.2 Methods

In this experiment, student participants worked in teams to design the first prototype of an app interface. We observed the design process and investigated the duration and their final delivered prototypes.

4.2.2.1 Experimental Design

Two type of teams were organized, a mixed type, composed of students from CS-related disciplines and other disciplines, such as humanities and psychology and a single type, composed of students from only CS-related discipline. We created 25 groups that were assigned to complete the design task: ten mixed teams and five single teams serving as control groups and ten mixed teams serving as experimental groups. Between-groups measures were used to gauge the effect of providing the supporting document, which included a table of data from a hypothetical scientist observations (see Appendix D). Groups in the controlled scenario did not receive supporting document, while groups in the experimental scenario did receive it.

4.2.2.2 Participants

Seventy-nine college students (68 male and 11 female) participated in this study for extra course credits. Three-quarters of the students came from Computer Science (CS) or Computer Science-related disciplines, such as Computer Engineering; the reminder included eight percent Humanities students and 17% from other disciplines such as Mechanical Engineering, Civil Engineering. Among the participating undergraduate students, 35% were seniors, 32% were juniors, 27% were sophomore, and 2% were freshmen; in addition, 4% of the participants were graduate students. We tended to recruit senior-level students because they potentially have more experience in software development, particularly those who are Computer Science students.

4.2.2.3 Materials

In the experiment, the interface that participants were required to design was for a citizen science app that involved collecting information on deer and deer-vehicle collisions (see Appendix E). People would use this app to report the location of deer on the side of the roads or deer-vehicle collisions. Animal-related car accidents have resulted in many deaths and injuries, particularly in Michigan. The data would provide information about the migration patterns of the deer population for scientists and identify dangerous sections of roadways for drivers. Based on the user story, participants were asked to generate and prototype the app flow interfaces in about an hour (see an example in Appendix F).

The idea of the app comes from a citizen science app that students developed in 2015. The supporting documents provided were faked according to the information provided by the scientist. It is a table including the features that the scientist may mostly need (see Appendix D).

After they had completed the design, each participant was given a self-report questionnaire (see Appendix G). The questionnaire included background information questions and three retrospective questions:

- Select the ideas that you have IMPLEMENTED in the final design of your app.
- Select the ideas that you have CONSIDERED, VOCALIZED to your teammates, but NOT ADOPTED in the final design of your app.
- Select the ideas that you have CONSIDERED, but DID NOT VOCALIZED to your teammates.

A number of ideas were listed below each of the preceding questions, such as location by map, number of deer, type of deer and so on. The data of the post-study questionnaires will be analyzed based on groups.

4.2.2.4 Procedure

In the experiment, each team member first signed an informed consent agreement. Teams were composed of three participants and were presented with a prototyped interface example drawn on paper, along with several pieces of blank paper and pens. The experimenter also distributed the user story of the app and instructed the participants to spend about three minutes reading it. The experimental groups received the supporting document, which the teams could use for reference. Next, the participants worked together to create and prototype the app flow screens on paper. After they had completed the design, each participant was given the self-report questionnaire. Participants were instructed to place a check by the ideas they had implemented, discussed or considered and to add other ideas if their ideas were not listed. Pizza and snacks were provided during the experiment. Apart from the extra course credit, this is a big incentive for students to participate the experiment. Considering that all teams are provided with food and drink in the same procedure, we assumed that this will not bias team performance including completion time. The task completion time for each team was recorded and the experimenter took notes on the ideas generated during the design process. The whole procedure was recorded by video.

4.2.2.5 Data Analysis

Three factors were measured in this experiment study: task completion time, task completeness (meeting the minimum app requirements), and the number of ideas generated. Task completion time was measured from when participants began to read the user story to when they completed prototyping. To measure completeness, we assigned six parameters as the minimum app requirements and checked how many of them each team had included in their design. If ideas other than the six minimum app parameters were generated, they were recorded and used to measure diversity. An between-subjects analysis of variance (ANOVA) was used to compare variance of completion time and completeness and diversity of generated ideas by the mixed teams without supporting document, mixed teams with supporting document, and single teams without supporting document. A post hoc test, the Tukey HSD (honest significant difference), was performed after each ANOVA test to determine which groups differed.

A t-test was performed to compare the mean differences of the responses to the three retrospective questions after the experiment between the two scenarios, with and without supporting document.

4.2.3 Results

Fig. 4.4 shows the distributions of task completion time for each type of team. The mixed teams with supporting document used less time (32 ± 4 minutes) than the other two groups without supporting document (42 ± 7 minutes for mixed teams and 44 ± 8 minutes for single teams). Table 4.5 shows that the amount of time it took each type of team to complete the interface design was significantly different, F(2,22)=7.3, $p = 0.0037$. The post hoc test clarifies that mixed teams without supporting document were significantly different from the other two types of teams (see Table 4.6).

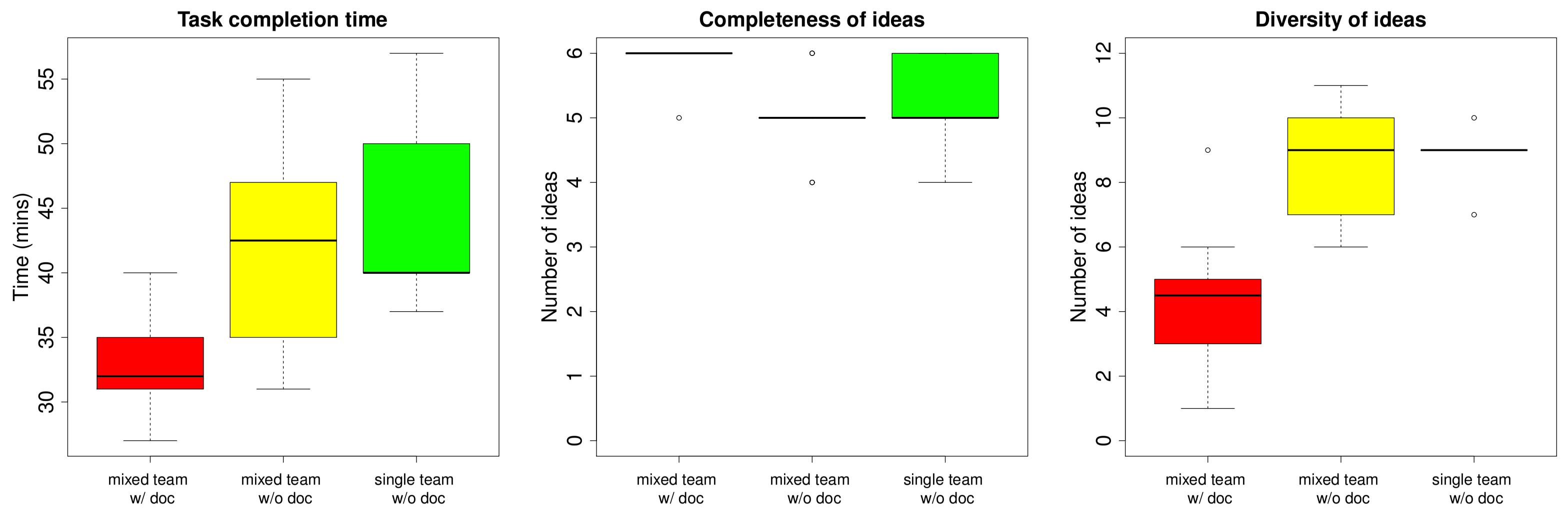

Figure 4.4 Comparison of time (left), completeness (middle), and idea diversity (right) among the three types of teams.

Table 4.5. ANOVA Analysis for Variance of Task Completion Time

	df	SS	MS	F value	p value
Between groups	2	655	327.5	7.3	0.0037*
Residuals	22	986	44.8		

Table 4.6. Tukey HSD Post Hoc Testing on Variance of Task Completion Time

	diff [a]	lwr [a]	upr [a]	q value [b]	p value
mixed team w/o doc - single team w/o doc	2.6	-6.6	11.8	1.00	0.76
mixed team w/o doc - mixed team w/ doc	-9.4	-16.9	-1.88	4.44	0.01*
single team w/o doc - mixed team w/ doc	-12.0	-21.2	-2.79	4.63	0.01*

[a] diff indicates the difference between means; lwr and upr are the lower and upper bounds of the 95% confidence interval.

[b] q value or studentized t value is based on a studentized range distribution with the number of populations of 3 and degree of freedom of 22 ($q_{0.05} = 3.55$).

The center graph in Fig. 4.4 shows the number of necessary features to complete of the app generated by each type of team. For the six features that constituted completeness (the left graph), almost all the mixed teams with supporting document generated all of the six features. The other two types of teams generated relatively fewer necessary features than the teams that received the supporting document. As shown in Table 4.7, there were significant differences in the number of completeness ideas among the three types of teams, F(2,22)=6.14, $p = 0.0076$. The post hoc test shows that results for mixed teams without supporting document were significantly different from mixed teams with supporting document (see Table 4.8).

Table 4.7. ANOVA Analysis for Variance of Completed Ideas

	df	SS	MS	F value	*p* value
Between groups	2	4.3	2.15	6.14	0.0076*
Residuals	22	7.7	0.35		

Table 4.8. Tukey HSD Post Hoc Testing on Variance of Completed Ideas

	diff [a]	lwr [a]	upr [a]	*q* value [b]	*p* value
mixed team w/o doc - single team w/o doc	0.2	-0.61	1.01	0.87	0.81
mixed team w/o doc - mixed team w/ doc	0.9	0.23	1.56	4.81	0.007*
single team w/o doc - mixed team w/ doc	0.7	-0.11	1.51	3.06	0.10

[a] diff indicates the difference between means; lwr and upr are the lower and upper bounds of the 95% confidence interval.

[b] q value or studentized t value is based on a studentized range distribution with the number of populations of 3 and degree of freedom of 22 ($q_{0.05} = 3.55$).

The graph on the right in Fig. 4.4 shows the distribution of diverse ideas for each team setting. The two teams without supporting document generated more varied ideas than the teams with supporting document. The results in Table 4.9 shows significant difference for the number of complete ideas among the three types of teams, F(2,22)=15.29, $p < .001$. As shown in Table 4.10, the post hoc test results for mixed teams without the document were significantly different from the mixed teams with the document. However, mixed teams without the document did not yield considerable more ideas than single discipline team without the document. This is contradictory to our hypothesis 3.

Table 4.9. ANOVA Analysis for Variance of Generated Ideas

	df	SS	MS	F value	p value
Between groups	2	107.6	53.78	15.29	6.89×10^{-5}*
Residuals	22	77.4	3.52		

Table 4.10. Tukey HSD Post Hoc Testing on Variance of Generated Ideas

	diff [a]	lwr [a]	upr [a]	q value [b]	p value
mixed team w/o doc - single team w/o doc	0.1	-2.5	2.7	1.38	0.99
mixed team w/o doc - single team w/ doc	-4.2	-6.3	-2.09	7.08	0.0001*
single team w/o doc - mixed team w/ doc	-4.3	-6.9	-1.7	5.92	0.001*

[a] diff indicates the difference between means; lwr and upr are the lower and upper bounds of the 95% confidence interval.

[b] q value or studentized t value is based on a studentized range distribution with the number of populations of 3 and degree of freedom of 22 ($q_{0.05} = 3.55$).

Participants were asked to recall the number of ideas they considered, discussed, and implemented during the development process; the number of ideas they considered and discussed but did not ultimately implement; and the number of ideas they considered but did not vocalize. We analyzed the responses to these three retrospective questions in the post- study survey. A t-test was performed to compare the number of ideas between the groups with and without supporting document. Table 4.11 shows the average number of ideas and variation for the three questions. Most ideas that were considered were articulated and discussed with other team members. The statistics show no difference between groups with supporting document and without supporting document. For the ideas that were considered, vocalized and implemented,

Table 4.11. Summary of the Self-reported Ideas from the Post-study Survey

	Considered, Vocalized and Implemented	Considered and Vocalized	Considered
	mean ± *sd*	*mean* ± *sd*	*mean* ± *sd*
w/o doc	8.67 ± 2.5	0.76 ± 0.8	0.49 ± 0.9
w/ doc	8.38 ± 2.0	0.88 ± 1.2	0.71 ± 1.4

the statistical result is $t(77)=0.55$, $p=0.58$; for the ideas that were considered and vocalized, the statistical result is $t(58)=-0.53$, $p=0.59$; for the ideas that were considered only, the statistical result is $t(52)=-0.77$, $p=0.43$.

4.2.4 Discussion

Overall, all the three measures (time, completeness, and diversity) presented supporting evidence to our hypotheses 1 and 2 that software teams, with supporting documents with an explicit table for requirements, spent less time and were more likely to generate all of the basic app requirements. But their ideas were restricted by the document, and they generated fewer ideas. Participants tended to discuss their ideas with their team members during the design process, rather than keeping ideas to themselves, regardless of whether or not supporting document was provided. However, the difference on diversity of app design ideas between mixed and single discipline teams is not significant as we hypothesized. The interdisciplinary collaboration is not a strength factor to influence the number of diverse ideas in the experiment. We attribute this to the characteristics of our teams in the experiment. The teams were temporarily established to complete a quick task within one hour. The collaboration was too short for team members to know each other and to be influenced by their disciplines. In the context of the course, one semester could be long enough to have such disciplinary impact.

Requirement engineering is heavily reliant on documentation as a way for stakeholders to communicate and share knowledge with software developers. Researchers suggest that a requirement specification document can even be part of the contract Paetsch et al. (2003). Our study showed the effectiveness of even documents that support requirement elicitation in improving team efficiency. However, there is one caveat: when provided these documents, developers are less likely to think outside the box and tend to generate fewer unique ideas. While conducting this study, we noticed that participants in the experimental groups with the supporting document were more likely to focus on the document after reading the user story. They tended to use the supporting document as a guideline for their interface design. This may explain why they could complete the task faster than other groups.

4.2.5 Summary of Study II

This experiment provided evidence of the impact of supporting documents on app development. The document facilitates the app interface design by reducing the total amount of time spent on design and by guiding developers to complete more of the app requirements, but it also leads to less variety of design ideas. This result further supports Study I's findings about the role of documents in requirement gathering and the entire software development process. On one hand, improving team efficiency is very beneficial in software development because software projects are usually very time sensitive, and schedule delays can result in cost and budget overruns. On the other hand, software development is a creative process, that should not be overly restricted by any technologies or tools.

4.3 Conclusion of the Study of the Collaboration between Development Teams and Clients

Hoegl Hoegl & Gemuenden (2001) indicated that teamwork quality alone cannot account entirely for team performance. Besides teamwork, many other factors can influence a team's performance, including management and organization, interaction between the team and other stakeholders, and team members' traits. In our studies, stakeholders other than developers, such as clients, play a more important role during the project development process than what we had expected. Development teams do not work on projects in isolation, but work within a stakeholder system in which everyone directly or indirectly relates and contributes to the project. Project managers and developers need to involve other stakeholders much more during the development process instead of working in an isolated environment, even though this will probably consume more time and lead to other issues.

Requirement gathering is the first phase in the software development process and is very critical because the process involves eliciting clients' needs and translating them into details of the software. Documents supporting requirement elicitation can effectively facilitate software development process by improving the scheduled progress performance, particularly in the requirement gathering phase. However, our experiment also indicates that, the documents confined the variety of ideas generated by developers. Participants focused on the supporting documents and were less likely to generate more creative ideas for the app. Therefore, rather than focusing only on documents, developers should communicate with clients in person frequently until reaching agreement on an app's requirements and should adapt their app solutions to correspond to clients' changing requirements. This also suggests that managers must get clients more involved in the development process, for example, by holding more meetings with development teams.

Chapter 5

Intra-Team Collaboration

Intra-teamwork quality can be measured by six aspects: communication, coordination, balance of member contributions, mutual support, effort, and cohesion (Hoegl & Gemuenden, 2001). [1] We investigated our teams on all these aspects through observation, questionnaires, and interviews, with an emphasis on the coordination and communication within teams. To achieve the common task goal of developing a citizen science app, team members need to coordinate, which means planning individual "parallel subtasks" to reach an agreement on work structures, schedules, budgets, and deliverables (Hoegl & Gemuenden, 2001). The aim of coordination is for each team member to be clear about their sub goals. In order to study team coordination, we used social network analysis to explore team information flows

[1] Part of the contents presented in this chapter have been published in the journal of ACM Transactions on Computing Education (TOCE) (Pastel et al., 2015).

and delineate team working structures.

Numerous studies (Gemuenden & Lechler, 1997; Griffin & Hauser, 1992; Hoegl & Gemuenden, 2001) support the theory that frequent communication within teams helps to insure project success. Our surveys placed an emphasis on studying team members' communication, in particular between team members from different disciplines, because we considered communication one of the most important measures of teamwork for our interdisciplinary teams. However, interdisciplinary communication is usually not easy and can cause many problems.

The first section describes a study on intra-team collaboration, focusing on the investigation of team structures and dynamic team mental models. To analyze intra-team collaborations, we closely observed the development teams in 2015, mainly to investigate team structure and team members' mental models.

The second section summarizes a further study on interdisciplinary collaboration and communication. Each year a survey was given at the end of the semester to collect individual perceptions on interdisciplinary collaboration. The courses were modified in both 2014 and 2015 regarding the interdisciplinary collaboration according to findings of previous year surveys, such as adding more lectures on teamwork. The details of the course modification and the comparisons before and after the modifications are introduced in the second section of this chapter.

5.1 Intra-Team Collaboration across Disciplines, 2015

According to the literature (Cooke et al., 2004; WIldman et al., 2012), a team's mental model represents not only the sum of individual conceptual knowledge, but also a knowledge structure from the team interaction process. Consequently, a network model, composed of relationship between individual team members, is used to investigate global properties of the teams (Avnet & Weigel, 2013). This network approach for studying teams is based on the principle that the shared knowledge of the team is "a synergistic functional aggregation of a team's mental functioning representing similarity, overlap, and complementarity" (Langan-Fox et al., 2004). It can offer a new perspective when studying team cognitive processes.

5.1.1 Hypothesis

Software teamwork quality and performance vary due to a number of influences, including team structure. We hypothesized that a team's structure will affect the team's mental model and performance during the development process. Because the study was conducted in the context of a course, most students are aware of and are trying to stick to the assignment schedule. From Study I, we learned that scientists' involvement can impact the development process. Consequently, the course instructors invested significant effort preparing the scientists before the beginning of each semester. The instructors met separately with each scientist to refine their concept of the app, to explain the development process, and to describe their role in the process. The instructor met multiple times with an individual scientist until he felt satisfied that the scientist had an appropriate concept of the app, had all readily available all the resources that the team would need to develop the app, and had an appropriate understand of their role in the development. Consequently, we are confident that the effect due to individual differences between scientists is modulated and that we can study the effect of team structure relatively independent of the effects of scientists. We assumed that if teams have different structures, the differences would mainly influence team performance on effectiveness, i.e. the quality of the software products.

As suggested from the responses to the open ended questions in the 2014 survey, leadership is important to the teams. Consequently, we also hypothesized that a team with effective leaders will demonstrate better performance than teams without good leadership. We will quantify team leadership by the centrality scores from the team's social network analysis and the ratings of team members' performance.

Hypothesis 1: Team working structures influence the effectiveness of software development teams.

Hypothesis 2: Teams with better leadership perform better.

5.1.2 Methods

We used two surveys: Cognitive social structure (CSS) survey and team dynamic process survey to investigate team structures and their potential impact on team performances. CSS survey is used to capture team structures. Because of the difficulty in measuring team performance directly, we used a team dynamic survey to measure performances. We compared the results of the two surveys to delineate how different team communication structures may influence the team performance.

5.1.2.1 Participants

All 35 students in the course's six teams completed the CSS surveys. The participants of the team dynamic process survey were the same as who did the cognitive social structure survey. However, not all the team dynamic process surveys were responded by all the students in the course, because some were absent when we distributed the survey. Considering that students's minds may change over time, we did not ask students to make up the survey if they missed it. Thirty-one students responded to the first survey, 35 to the second survey and 29 to the third survey.

5.1.2.2 Materials

Cognitive Social Structures (CSS) Survey. Cognitive social structure (CSS) surveys are tools used to collect data for social network analysis on teams. The surveys focus on collecting team members' perception of the relationship among themselves, and the network represents these relationships (Krackhardt, 1987). Mueller & Elizabeth (2008) used the social network data from Krackhardt (1987) to examine their Cultural Mixture Modeling for identifying cultural consensus. They found two distinct groups: the hierarchical group who believed that advice was through a few high-level managers and presidents; the democratic group (composed of a few managers) believed that advice was sought equally and distributively. Therefore they suggested that team members can better understand how the team operated than central managers. This corresponds to the value of the agile development, that it is not the leaders solve problems, rather than teams do. Recently, Brands (2013) made a review highlighting the application of cognitive social structures in social network analysis. The review investigated two questions that CSS research studies. 1) How do team members perceive their relationships? and 2) How do these perceived relationships and networks affect their behaviors and team outcomes? Brands found that significantly more studies pursued the first question, but the second question is currently more interesting. In our study, we applied the CSS survey developed by Krackhardt (1987) and explored how the team structures affected the team outcomes. For an example, Emily is a team member. Each team member, including Emily, was asked who Emily would go to for help during the software development process. To respond, participants chose from a list of all team members. The respondent can choose as many team members as they think appropriate. The question was repeated for each team member.

This study extends standard CSS techniques to studying software development teams. Our CSS survey is based on(Krackhardt, 1987) and asks two questions about who goes to whom for help and who shares ideas with whom (see Appendix H). Participants were asked to choose who they are likely to go to for help if they encounter problems or have issues. And if they have new ideas, who do they want to share with.

The Team Dynamic Process Survey. In previous studies, we used the usability test reports produced by the graduate students in the HCI course to measure team performance. Software errors and usability ratings were used as measures. The questions used for rating usability were, "do you think the app is easy and intuitive to use?" However, we were concerned about differences in usability tests, administrators and the small sample size of the tests (3 to 7 participants for each app). In order to investigate team performance from another perspective and to explore the team's mental models, we designed a team process survey to probe team members' attitudes, feelings and other mental attributes. The survey investigated individual perceptions on the project, collaboration and performance of other team members and so on. The questions required either open-ended responses or responses on a Likert scale.

We focused on five rating questions and two descriptive questions in the survey:

- Rate your satisfaction of you app.
- Rate your enjoyment in working with your teammates.
- Rate the severity of your team conflict.
- Rate the average CS team members' performance.
- Rate the average HU team members' performance.
- What is your goal for the project?
- What is the conflict in your team, if there is one?

5.1.2.3 Procedure

We distributed the team dynamic process survey three times in the 2015 spring semester, corresponding to conclusion of key phases in the development process (See Appendix I). The first survey was given after requirement gathering on January 27, 2015. The second survey was given on February 23, after the design review. The third survey was given after usability testing at the end of the semester on April 15. In the second time, along with the team process survey, we also distributed the CSS survey on February 23, 2015. All the surveys were collected in paper.

5.1.2.4 Data Analysis

Cognitive Social Structures Survey. As explained in Krackhardt (1987), cognitive social structures are presented as a set of relationships derived from the survey responses. The data we collected from the CSS survey is a three-dimensional data set, a set of relationship matrices, $R_{i,j,k}$, where k is the perceiver (the person who completed the survey), i is the sender of the relationship and j is the receiver of the relationship. Thus $R_{1,2,3}$ would be interpreted as: team member 3 perceives that person 1 would go to person 2 for help when having problems during the development process.

In order to look into team structures from a social network analysis perspective, Krackhardt (1987) proposed three approaches to aggregating the three-dimensional data set into two dimensions. One aggregation technique is to hold the perceiver dimension k constant; which Krackhardt (1987) called this aggregation a *slice*. The second approach is to make a diagonal slice, meaning $k = i$ or $k = j$, among all the perceived relationships between i and j. Generally only those relationships that were provided by either i or j themselves are considered valid. This is called *locally aggregated structures* (LAS). The third approach averages the relationships between i and j as a across of all-perceivers. We then use a threshold between 0 and 1 to determine if the relationship exists. This is called a *consensus structure*.

From the CSS survey, we extracted several network structures for each team in 2015:

two slices, one LAS and one consensus structure. We used both questions (who will you go to for help and who will you share ideas with).As for the slice aggregation, we did not extract individual team members perceptions, instead we sliced the entire network into two parts: the CS team members and HU team members' perceptions. This allows us to compare perceptions between CS and HU team members. We used the union rule to aggregate the LAS, meaning a relationship exists if it is perceived either by the sender or by the receiver. Using the consensus structure, the relationships between i and j perceived by all team members were averaged, and a 50% threshold was used to make the social network.

The Team Dynamic Process Survey An ANOVA analysis was performed to determine whether there were differences between groups and between the three surveys (which were conducted at different times). A post hoc Tukey HSD test was performed after the ANOVA analysis to determine which groups and surveys differed. The two descriptive responses were also analyzed. At the beginning of the survey we asked team members to give their goal for the project. After rating the severity of team conflict, participants were asked to describe the conflict.

5.1.2.5 Results

Social Network Graphs. Figure 5.1 presents the team structures extracted from the first question (who would you go to for help), allowing us to compare perceptions between different aggregations and teams. In all of the network figures, CS1 to CS4 are CS team members; and HU1 and HU2 are HU team members. The CoCo team has only five team members. We can see that the structures are more different among teams than aggregations. There are some differences among the three aggregations.

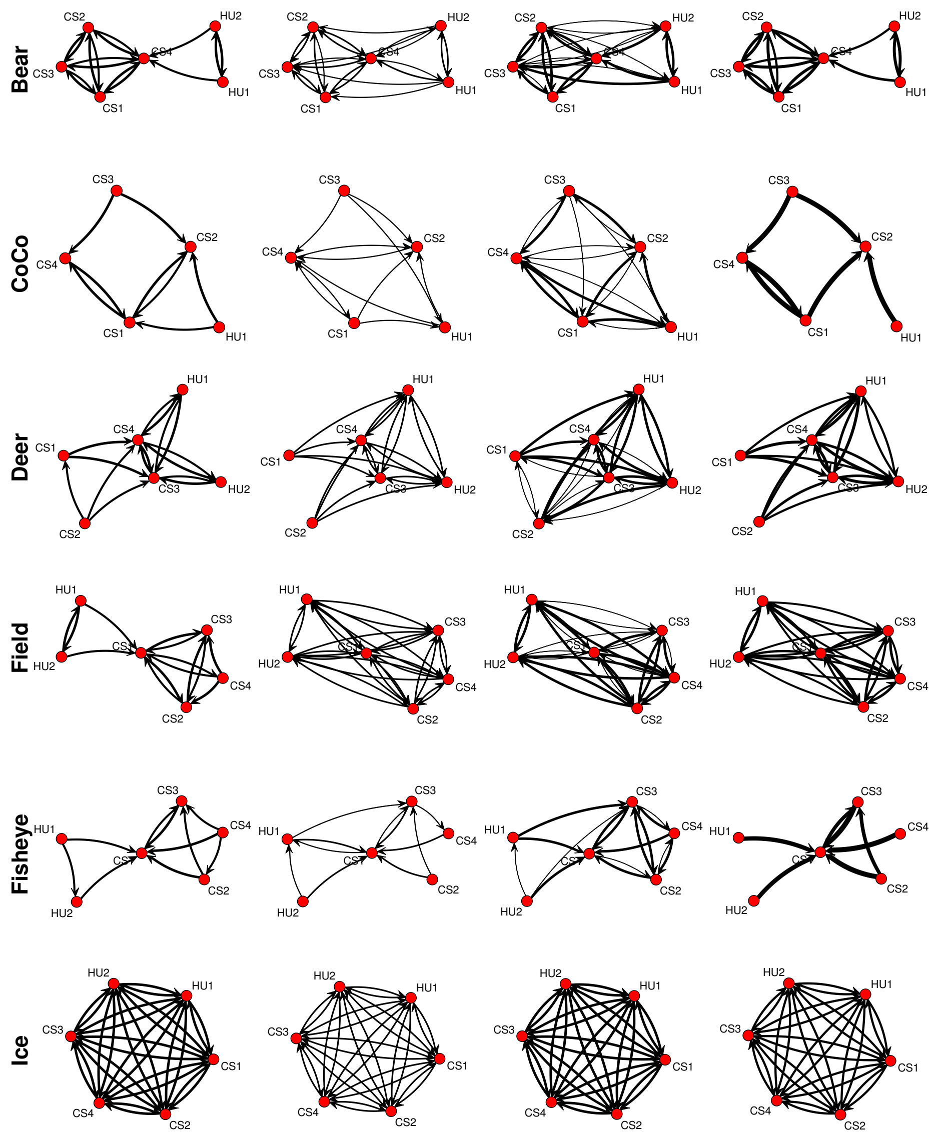

Figure 5.1 Social Network Structures of Development Teams in 2015. Each row indicates a team. Each column indicates an aggregating approach. The aggregations are (from left to right) CS slice, HU slice, LAS and Consensus.

The first two columns are CS slices and HU slices. In each team, the team members from both disciplines agreed on the general team structures. For example, the Bear, Deer, Field, and Fisheye teams are more centralized teams, with one or two central members, while the Ice team has a fully connected structure, CoCo team has less connections. In the centralized teams, the CS team members are more likely to be in the center, while the HU students are often peripheral team members, except in the Deer team. In the Deer team, two CS team members and two HU team members are in the center.

However, there were differences between CS and HU team members' perceptions. The CS team members thought the HU team members were not as helpful as CS team members, because in most teams CS members would not go to HU1 or HU2 for help (in team Bear, CoCo, Deer, Field and Fisheye, See the first column in Figure 5.1). From CS members' perspective, HU members only communicated with their HU teammate and the central person. However, the HU team members perceived themselves as more central and communicating with more CS team members. As the second column in Figure 5.1 shows, on the Bear team, HU1 and HU2 would go to CS members for help rather than only communicating with central person (CS 4). In the CoCo team, the HU member thought all CS members would go to him for help. Therefore, HU team members tended to believe that they communicated with CS members more than was perceived by the CS members.

The third column of Figure 5.1 presents the LAS aggregated structures. Three teams showed fully connected structures: Deer, Field, and Ice; the others are relatively central or hierarchical. We can also see some asymmetric relationships. For example, in the Fisheye team, CS5 only goes to CS1 and CS3 for help. Person 6 only goes to CS1 for help. However, no one goes to HU1 and HU2 for help. The centralized teams have one or more "help" centers. These team members were perceived by the rest of the team as the person to go for help. Examples of "help" centers are CS1 in Fisheye and CS4 in Bear. Consequently the central team members are always the CS members and the HU members are more peripheral.

The consensus method showed in the last column indicates more connections between team members. The structures of team Bear, CoCo and Fisheye are similar to the CS slices, the first column of Figure 5.1. With the threshold, the Bear and Fisheye teams are very centralized with one strong, central person (CS4 in Bear and CS1 in Fisheye). The Field and Ice teams are fully connected without apparent centers. The Deer team is neither centralized nor fully connected, rather it has a subcomponent that is fully connected and two members that seek help only from the fully connected subcomponent and no one seeking help from them. The subcomponent is centralized around more than one person, and the central persons are not only CS students.

In general, the four aggregations indicate similar structures but slight difference. The CS slice shows that CS students tend to believe the teams are centered by

CS disciplines and they would not go to HU students for help. While the other methods suggest the HU members in the team Deer and Field are perceived to be more communicated.

Centrality Scores. To visualize the qualitative observations, centrality scores were calculated for each team in each of the three aggregations. The scores we measured were: indegree, outdegree and betweenness.

In 1954, degree was first introduced to be used as an index of centrality (Shaw, 1954). *Degree* is the count of the actors' connections in a social network. *Indegree* is the number of connections coming to an actor, and *outdegree* is the number of connections going out from an actor. L. C. Freeman (1979) explains that a relatively high degree is "in the thick of things" with respect to communication. The actor with higher degrees is likely to be seen as a major channel of information flow, thus the focal point of communication. Anthonisse (1971) and L. C. Freeman (1977) developed betweenness as another centrality measurement. *Betweenness* indicates the situation in which an actor is located in the between two other actors. The actor between the other two actors controls the communication because the other two actors have to rely on him or her in order to communicate. L. C. Freeman (1979) indicated a more complicated situation when more than one person connects a pair of team members. This partial betweenness was calculated in terms of probabilities. In this study, due to the small team size, we only included the situation when one person connects a

Table 5.1. Team Centrality Scores

Team	Subject	CS Slice		HU Slice		LAS		Consensus	
		degree	betweenness	degree	betweenness	degree	betweenness	degree	betweenness
Bear	1	29	0	12	0.25	11	0	41	0
	2	29	0	12	0.25	16	1.83	41	0
	3	30	0	13	0.5	15	1.83	43	0
	4	32	0	16	6.5	16	1	48	0
	5	25	0	9	0.25	11	0	34	0
	6	23	0	8	0.25	11	0.33	31	0
	Total	168	0	70	8	80	5	238	0
CoCo *	1	25.2	0.4	4.8	2.4	13.2	0.4	30	0.4
	2	24	0.4	6	0	12	1.2	30	0.4
	3	18	0.4	3.6	0	9.6	0.4	21.6	0.4
	4	24	0.8	7.2	4.8	14.4	1.6	31.2	0.8
	5	16.8	0.4	4.8	0	10.8	0	21.6	0.4
	Total	108	2.4	26.4	7.2	60	3.6	134.4	2.4
Deer	1	24	0	8	0	11	0	32	0
	2	21	0	8	0	14	1.5	29	0
	3	33	0	16	0	18	1.5	49	0
	4	33	0	16	0	15	0	49	0
	5	24	0	16	0	16	0	40	0
	6	25	0	16	0	16	0	41	0
	Total	160	0	80	0	90	3	240	0
Field	1	30	0	20	0	13	0	50	0
	2	28	0	20	0	19	0	48	0
	3	28	0	20	0	14	0	48	0
	4	26	0	20	0	18	0	46	0
	5	25	0	20	0	16	0	45	0

pair of team members. The pair of team members always has to rely on the team members in between to connect them.

Table 5.1 (cont'd)

Team	Subject	CS Slice		HU Slice		LAS		Consensus	
		degree	betweenness	degree	betweenness	degree	betweenness	degree	betweenness
	6	25	0	20	0	6	0	45	0
	Total	162	0	120	0	96	0	282	0
Fisheye	1	24	2	12	5	13	1	36	8
	2	16	0	3	0	10	0.5	19	0
	3	19	2	6	4	14	3.5	25	2
	4	15	0	3	0	8	0	18	0
	5	9	0	5	1	5	0	14	4
	6	9	0	3	0	4	0	12	0
	Total	92	4	32	10	54	5	124	14
Ice	1	40	0	20	0	20	0	60	0
	2	40	0	20	0	20	0	60	0
	3	40	0	20	0	20	0	60	0
	4	40	0	20	0	20	0	60	0
	5	40	0	20	0	20	0	60	0
	6	40	0	20	0	20	0	60	0
	Total	240	0	120	0	120	0	360	0

*The scores for CoCo team are scaled by 6/5, because CoCo team has only five members.

Table 5.1 shows the degrees and betweenness of each team for the three aggregation approaches. In general, the Fisheye and CoCo teams have lower degrees than other teams in all the structures. This indicates that from the perspective of the team members, the teams Fisheye and CoCo have less communication channels. As for betweenness, Fisheye, CoCo and Bear teams have members perceived as serving a central role in the team.

The Pearson correlation coefficient was computed between each pair of aggregation methods for each team. Table 5.2 shows relatively high correlation among all the methods. CS slices and Consensus method are especially highly correlated, because CS students represent the majority of the team members. Some of the correlation coefficients from HU slice and the LAS method are relatively low compared to those from CS slice and Consensus. We attributed it to the small sample size of the HU

Table 5.2. Team Matrix of Correlation Coefficient

Team	Subgroup	Degree				Betweenness			
		CS Slice	HU Slice	LAS	Consensus	CS Slice	HU Slice	LAS	Consensus
Bear	CS Slice	1	0.98	0.76	1.00	1	-	-	-
	HU Slice	0.98	1	0.77	0.99	-	1	0.12	-
	LAS	0.76	0.77	1	0.77	-	0.12	1	-
	Consensus	1.00	0.99	0.77	1	-	-	-	1
CoCo	CS Slice	1	0.61	0.83	0.97	1	0.88	0.75	1.00
	HU Slice	0.61	1	0.83	0.77	0.88	1	0.61	0.88
	LAS	0.83	0.83	1	0.90	0.75	0.61	1	0.75
	Consensus	0.97	0.77	0.90	1	1.00	0.88	0.75	1
Deer	CS Slice	1	0.63	0.52	0.92	1	-	-	-
	HU Slice	0.63	1	0.82	0.88	-	1	-	-
	LAS	0.52	0.82	1	0.72	-	-	1	-
	Consensus	0.92	0.88	0.72	1	-	-	-	1
Field	CS Slice	1	-	-0.44	1.00	1	-	-	-
	HU Slice	-	1	-	-	-	1	-	-
	LAS	-0.44	-	1	-0.44	-	-	1	-
	Consensus	1.00	-	-0.44	1	-	-	-	1
Fisheye	CS Slice	1	0.75	0.93	0.96	1	0.98	0.80	0.64
	HU Slice	0.75	1	0.60	0.90	0.98	1	0.69	0.79
	LAS	0.93	0.60	1	0.86	0.80	0.69	1	0.15
	Consensus	0.96	0.90	0.86	1	0.64	0.79	0.15	1
Ice	CS Slice	1	-	-	-	1	-	-	-
	HU Slice	-	1	-	-	-	1	-	-
	LAS	-	-	1	-	-	-	1	-
	Consensus	-	-	-	1	-	-	-	1

*NA is because one or more subgroups had the same degree/betweenness score for all subjects in a team. No correlation coefficient is computed in this case.

team members and in the LAS method.

Summary of Cognitive Social Structure Survey. CS and HU team members' cognitive social structures were compared in social network analysis graphs. Team members were in agreement about the general structure of their team. However, there were differences between how the CS and HU members perceived the connectedness of HU members in the team. HU students thought they were more central and communicated with more team members than their CS team members thought. From a consensus aggregation perspective, the Deer team showed moderate centralization with more than one central person. The Fisheye, Bear and CoCo teams are centralized and only CS team members are centers. The Ice and Field teams are fully connected.

Does team structure differences affect team performance? A study of software product development teams indicated that the centralization of a team network was negatively related with creative performance of the team (Leeders et al., 2003). However, in our case, as suggested by previous teams and from students' responses in the surveys, leadership is needed. Therefore, we wanted to study the influence of these different team structures.

The Team Dynamic Process Survey. Table 5.3 shows the ANOVA analysis of the ratings between teams on the five rating questions. Responses to four out of the five questions appeared to be significantly different among the teams. Ratings on CS team members' performance were also different, although not statistically significant ($p = 0.05$). Table 5.4 indicates which pairs of team differs when a post hoc Tukey HSD test was performed. (Results with p -values less than .1 are listed in the table.) For all questions, the Deer team rated themselves higher than either the Field or the Fisheye team. The Ice team also had higher ratings for HU team members' performance than the Field or the Fisheye team.

We also analyzed the rating changes over time for each team. No specific trend in the changes was observed, but only the Ice team had a significantly decreasing ratings

Table 5.3. ANOVA Results for Ratings Responses in Dynamic Team Process Survey

Question		df	sum of sq	mean of sq	F value	p value
Satisfaction	Between teams	4	20.59	5.15	4.33	0.01*
	Residuals	73	86.76	1.19		
Working Enjoyment	Between teams	4	20.15	5.04	3.81	0.01*
	Residuals	73	96.53	1.32		
Conflict	Between teams	4	38.46	9.62	3.22	0.01*
	Residuals	73	218	2.99		
CS Performance	Between teams	4	16.39	4.10	2.44	0.05
	Residuals	73	122.79	1.68		
HU Performance	Between teams	4	15.11	3.78	5.07	0.01*
	Residuals	73	54.43	0.75		

Table 5.4. Tukey HSD Post Hoc Testing on Variance of Ratings in Dynamic Team Process Survey

Question	Teams	Mean Difference	p value
Satisfaction	Deer > Field	1.19	0.03*
	Deer > Fisheye	1.41	0.01*
Working Enjoyment	Deer > Field	1.47	0.01*
	Deer > Fisheye	1.08	0.06
Conflict	Deer > Field	1.67	0.08
	Deer > Fisheye	1.97	0.01*
CS performance	Deer > Field	1.39	0.04*
HU performance	Deer > Field	0.86	0.07
	Deer > Fisheye	0.91	0.03*
	Ice > Field	1.08	0.02*
	Ice > Fisheye	1.13	0.01*

over all five questions. Next, we studied the descriptive responses in the survey. Figure 5.2 shows the responses to the project goal question for three times at which the survey was given. Percentages are given for each goal, indicating the percent of the respondents that thought a particular goal was important. The figure indicates that there were three main goals from project beginning to end: creating an app, completing functionality, and making the app easy to use. The percentages of two other goals, meeting clients' requirements and completing the app, increased over time.

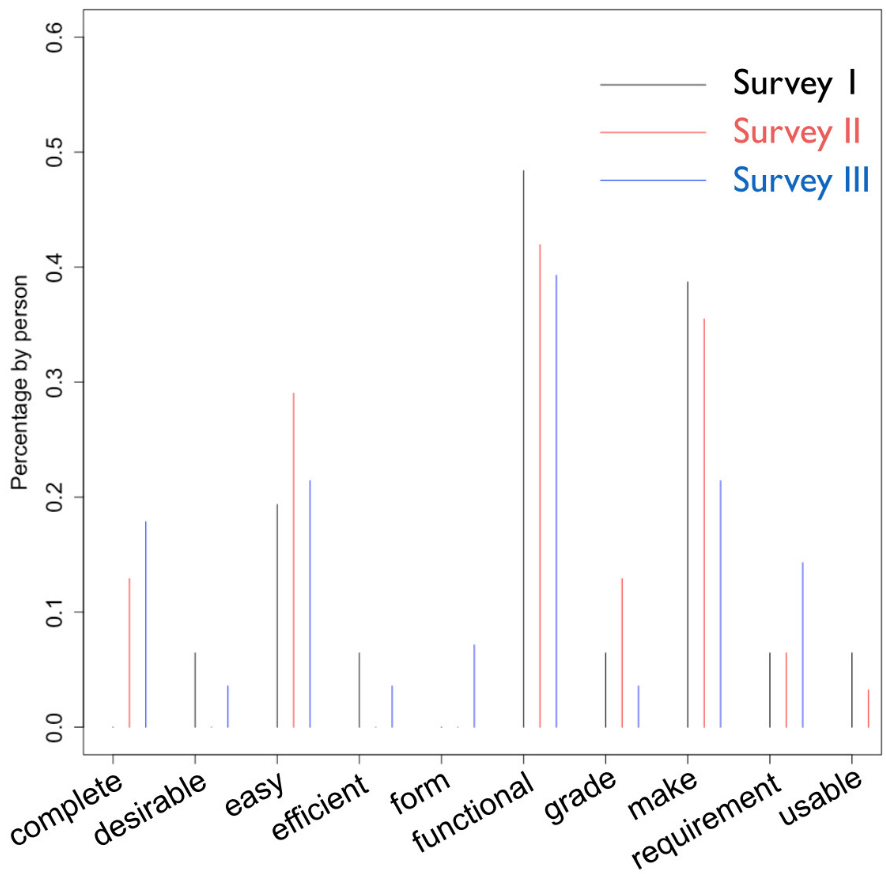

Figure 5.2 The descriptive responses on project goals as they changed over time.

Figure 5.3 shows a comparison of responses between CS and HU team members. The two disciplines shared similar goals, except that HU members cared about their grades more than CS members.

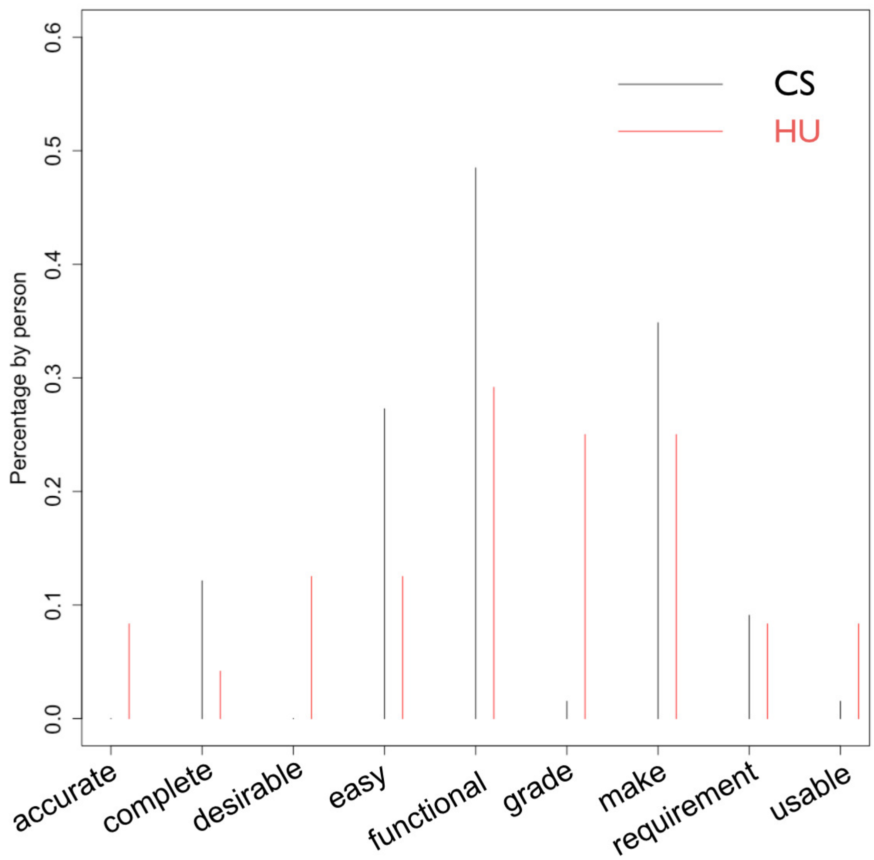

Figure 5.3 Comparison of the descriptive responses on project goals between CS and HU disciplines.

To sum up the results from the dynamic surveys, the Deer team has higher ratings than the Field and Fisheye teams in general. Only the Ice team's rating changes over time, decreasing as development proceeded. We did not observe the ratings change with time for the other teams. The descriptive responses showed that except for the three major goals, creating an app, completing the functionality and making the app easy to use, students' goals changed to completing the app and getting a good grade. HU team members cared more about grades.

We also summarized team conflicts from the participants' survey responses. Each team had complaints about small conflicts in communication from one or two team members. Some of these issues were resolved during the development process, according to team members. Only the Field team had unresolved conflicts with their scientist. Several students, from both CS and HU, complained that their scientist changed the requirements for the app, particularly near the end of the semester. These requirement changes cause the team to rush the new design and implementation of the app at the end of the project.

Summary of the Team Dynamic Process Survey. The dynamic survey tells a different story about these teams. Table 5.5 summarizes the results of the rating questions on app satisfaction, working enjoyment, conflict and team members performance. The team with higher ratings is Deer team, which is moderately centralized and with connected members from both disciplines. Because we did not find obvious trends in the change of ratings over time for most of the teams, we studied the descriptive responses on project goals and conflict might for more information. The goals changed with time in several aspects across all participants. While app creation, functionality, and ease of use remained important, more people aimed at completing their projects and getting a good grade as the semester went on. The descriptions of conflicts reveal that the Field team had problems with their scientist in requirement gathering, especially near the end of the project.

Table 5.5. Ratings Summary According to Teams

Team	Structure	Ratings	Change over time	Central people	Other issues
Deer	moderately centralized	high	no change	CS & HU	N/A
Fisheye	centralized	low	no change	CS	changing leadership
Field	fully connected	low	no change	N/A	client
Ice	fully connected	high	decreased	N/A	N/A
CoCo	centralized	moderate	no change	CS	N/A
Bear	centralized	moderate	no change	CS	N/A

5.1.3 Conclusion of the Intra-Team Study

The six teams had different working structures. The Fisheye team is the most centralized team and the Bear, CoCo and Deer teams are moderately centralized. Among the three teams, members from different disciplines only mingled well on the Deer team. The other two teams, the Bear and CoCo team, showed divisions between members of the CS and HU disciplines. The Ice and Field teams were relatively connected between team members. According to the dynamic team process surveys, in general the Deer team had better satisfaction, enjoyment, conflict, and performance ratings than the Field and Fisheye teams. From our close observation of these teams, the Deer team was highly functional with involvement from both disciplines, effective leadership by both disciplines, and participation from the rest of the team members. On the other hand, the Fisheye team is centralized and had a team leader. (We learned from observation that they changed leaders during the development process.) The second leader was a strong person that could lead and take over the project. However, a team project may need more collaboration among members rather than one strong individual coordinating all the tasks and do most of the implementation.

We believe that the Field team's problems were brought about by the changing requirements from the team's client. In the previous study, we learned that, although clients are separate from the development teams, they can have an important role. Failing to provide development teams with information, such as timely requirements, can lead to project delays, product incompleteness, and even project failure. The Field team case provided further evidence for what we learned about the importance of clients.

5.2 Interdisciplinary Collaboration, 2013-2015

In this section, we focus on the collaboration between the undergraduate students in the CS and HU courses (i.e. software engineers and technical communicators). We investigated the collaboration between these two stakeholder groups to understand the interdisciplinary collaboration process in software development by peer evaluation and observing a team shared mental model. The aim was to delineate collaboration issues and seek approaches to resolving these issues and improving the process.

The two groups of stakeholders within the development teams were studied: software engineers and technical communicators. The boom of computer and Internet usage has increased the role of technical communication experts (O'Hara, 2001) in software development. Technical communicators identify and effectively communicate information to the end-users in these products. Typically, they write product descriptions, tutorials and documentations according to users' needs, in order to effectively communicate the information implemented by software engineers. Technical communicators also conduct usability evaluations on the communication and the product.

5.2.1 Methods

To understand the collaboration between disciplines, we conducted surveys and semi-structured interviews at the end of each semester from 2013 to 2015. As previously mentioned, communication is usually considered a component of collaboration, and over time communication within a team can become the collaboration (Jackson, 2010). Consequently, the end of semester surveys during all three years asked about communication between the two stakeholder groups. Apart from the surveys, we modified the courses according to the insights from the survey results, such as coordinate between two course instructors, giving agreement between scientists and development teams etc. They survey is used for evaluate the effect of the modifications and getting hints of potential issues.

5.2.1.1 Participants

Nineteen CS students and 8 HU students responded to the survey in 2013. 27 CS students and 10 HU students responded to the survey in 2014. 23 CS students and 9 HU students responded to the survey in 2015. In the semi-structured interviews, a focus group discussion with at least one member from each team, and we interviewed with the entire team all three years.

5.2.1.2 Materials

In the surveys at the end of the semester, students were asked to rate the communication performance of teammates in each year (See Appendix A and Appendix B). In 2014, the survey additionally asked detailed questions about the teams' shared mental model and team conflicts. In the interviews, we asked students about the development process and probed with more detailed questions, e.g., "how do you arrange meetings?" or "is it hard to get help from the members from the other discipline?".

5.2.1.3 Procedure

This process took about three years. At the end of each spring semester from 2013 to 2015, we distributed a survey to evaluate the communication process and team performance. The interviews with teams and individual team members at the end of each semester helped to explore more information that the survey can not provide. Regarding the insights from the surveys and interviews each year, we modified the course each year such as adjusting the schedules and education teams.

5.2.1.4 Data Analysis

T test is mainly used for comparing the survey results of different years. For the 2014 survey, we also compared the team members mental models on the important aspects of app and possible influential factors to project. The percentages were calculated and compared between different years and team members from different disciplines.

5.2.2 Results

Table 5.6 summarizes and compares the communication ratings of developers' own discipline and the other discipline from 2013 to 2015. The ratings were from one to

Table 5.6. The Communication Performance Ratings from Developers on Their Team Members within Discipline and from the Other Discipline

		Within discipline *mean* ± *sd*	With other discipline *mean* ± *sd*	*t*-value	df	*p*-value
2013	CS n=19	5.31 ± 1.1	4.07 ± 1.6	2.70	29	0.01*
	HU n=8	5.84 ± 1.4	3.65 ± 1.6	2.93	14	0.01*
2014	CS n=27	5.85 ± 1.1	4.33 ± 1.6	3.81	51	0.0004*
	HU n=10	5.80 ± 1.6	4.00 ± 2.1	2.10	17	0.04*
2015	CS n=23	5.09 ± 1.4	5.70 ± 1.2	-1.58	43	0.12
	HU n=9	5.67 ± 1.7	5.67 ± 0.7	0	11	1

seven with one anchored at "very poor" and seven anchored at "very good."

The results from the 2013 survey show that all the ratings were above average. CS and HU students' ratings within their disciplines are not significantly different (t (11) = 0.97, p = 0.36). However, the communication ratings within their own discipline are significantly higher than the ratings of communication with the other discipline (CS with HU students: t (29) = 2.70, p = 0.01; HU with CS students: t (14) = 2.93, p = 0 .01). The communication with the other discipline is not rated as highly as within the same discipline. An effective interdisciplinary team should have equally effective communication across disciplines.

The collaboration between the technical communicators and software engineers was not as strong as hoped for in first year, 2013. Consequently, the instructors made four major modifications to the course collaboration in 2014 to improve the collaboration within the teams. To improve communication within teams, students from the two classes were given more time to meet during the class hour by arranging for the two courses to meet in adjacent classrooms. The second modification was to ensure that HU and CS students had shared goals by adjusting and scheduling the course assignments. The third modification attempted to better define the roles and commitment of stakeholders by distributing a document that described the roles of each stakeholder in the development process. In addition, the instructors asked teams to write a contract and for all team members to sign the contract. The fourth modification attempted to improve the shared mental model and common language of the HU and CS students by having one of the instructors give three lectures that were attended by both HU and CS students in the same classroom.

After modifying the course collaboration in 2014, we again had teams self-rate communication (see Appendix B). Unfortunately, the communication ratings of 2014 failed to show any significant improvement from year 2013 either within the same discipline (CS students: t (37) = 1.7, p = .10; HU students: t (16) = 0.05, p = 0.96) or for communication with the other discipline (CS students: t (39) = 0.51, p = .61; HU students: t (16) = 0.42, p = .69). According to the communication ratings the collaboration did not improve.

Anticipating the possibility that the collaboration might not improve, we studied the team mental models at the end of the 2014 semester (see Appendix B). Team members were asked to delineate the influences on development. The survey listed eight potential influences:

- Team collaboration
- Communication with clients
- CS team members' skills
- HU team members' skills
- Time to complete the app
- Team members' personalities

The question asked, "What influenced your app development?" Students could choose more than one influence. Table 5.7 shows the specific number of responses and percentages. Three out of the eight influences were most often chosen: team collaboration, communication with clients and CS team members' skills. An interesting result is that even among HU students only 10% of the respondents considered that the HU students' skills were influential. Also, while 32% of CS students responded that "time to complete the app" was an influence, only 10% of the HU students felt that time was a factor.

The next question on the survey asked developers to identify what had the biggest influence on their development process. They could choose only one influence in this

Table 5.7. What Team Members Thought Influenced the Process

Influence	CS	HU
Team collaboration	14(50%)	3(30%)
Communication with clients	15 (54%)	2 (20%)
CS team members' skills	15 (54%)	4 (40%)
HU team members' skills	3(11%)	1(10%)
Time to complete the app	9 (32%)	1(10%)
Team members' personalities	4(14%)	1(10%)

Table 5.8. What Team Members Thought Most Influenced the Process

Influence	CS	HU
Team collaboration	10(37%)	2(20%)
Communication with clients	7 (26%)	3 (30%)
CS team members' skills	8 (29%)	2 (20%)
HU team members' skills	0(0%)	1(10%)
Time to complete the app	1 (4%)	0(0%)
Others	1 (4%)	2 (20%)

question. Table 5.8 indicates the distribution of participants' choices in each discipline. Again, team collaboration, communication with clients and CS team members' skills were chosen as the most influential aspects of app development. A Pearson's Chi-Squared test does not show significant difference between the CS and HU distributions (X2(2, N=6) =3.28, p=0.66). There is a slight difference in that one CS student considered the time of the development to be critical and one HU students considered the HU members' skills to be critical to the app development.

In addition, developers were asked to select the aspects of their completed app that satisfied them and dissatisfied them. The developers could choose more than one aspect of the app:

- **Completion** of the app
- Convering all the **requirements** of the app
- **Appearance** of the app
- Correct **Functionality** of the app
- **Usability** of the app
- Complete help **documentation** for the app

Figure 5.4 summarizes the responses from student developers from the two disciplines. Generally speaking, students from the two disciplines were comparably satisfied and dissatisfied with covering all clients' requirements and completing the app. Student developers tended to be more concerned about the aspect of the app that they worked on. They were more likely to choose the aspects related to their own work compared to the work of others, either more satisfied or more dissatisfied with. HU students cared more about the app documentation and usability of the apps and were more dissatisfied with the appearance of the apps. The CS students were more concerned about the app having the correct functionality; they were either more satisfied or dissatisfied with their app's functionality than the HU students.

We examined the responses to open-ended questions in 2014 surveys and interview questions. We asked the developers to state if there were any team conflicts and

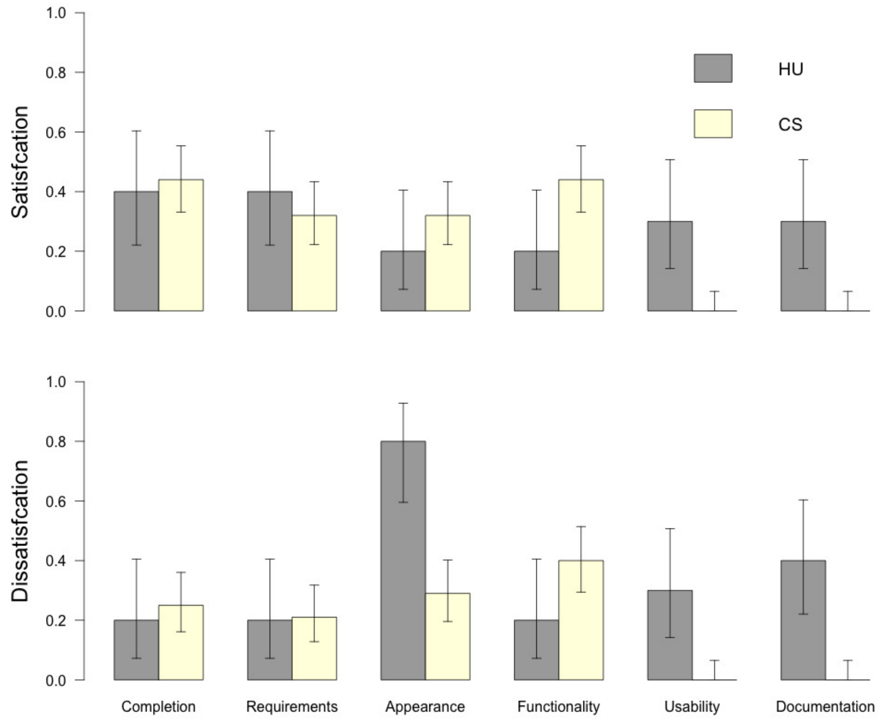

Figure 5.4 Percentage distribution of the aspects of the completed app that student developers were satisfied/dissatisfied with.

to describe the conflicts. Conflicts in a team usually are one of two types, task or interpersonal conflicts. Task conflicts refer to disagreement in opinions and ideas about how team tasks should be performed; interpersonal conflicts refer to disagreements and clashes in individual relations, typically involving negative emotions such as stress, frustration, and anger (Yong et al., 2014). Only HU students expressed interpersonal concern, which is the clash of personalities between the two disciplinary groups. Our summary of responses to open-ended questions in surveys and interviews found primarily task conflicts. The task conflicts reflected conflicts about communication, coordination and workload. Both the HU and CS students complained about communication with the other discipline. HU students blamed CS students for not communicating well or even not wanting to communicate. For example, some HU developers said, “CS students would not come for meetings” and “would not speak in the meetings”. The HU team members also wanted to get more involved in the development process, e.g. “we should keep CS and HU students on the same page” and “we should understand each others’ assignments better.” Some CS team members mentioned conflicts about the interface design between the HU and CS developers. Some CS team members described difficulties in getting the help content from the HU team members. Students from both HU and CS disciplines showed concerns on coordination, e.g. “need to develop more defined roles,” “lacking of a team leader who can assign tasks,” and “need to split the work more evenly.” At least one team expressed obvious regret in not having a team leader and attributed that as the reason for their failure to complete the app. HU team members expressed the desire for a clear expectation of what should be in the app’s documentation. In addition, some HU students also complained, “CS students didn’t see the help documentation as important”, which was to be a major contribution of the HU students to the project. Workload was a critical concern for CS team members. Many CS students expressed the need for more time to program and implementing the apps. No HU student shared this belief.

5.2.3 Discussion

The interdisciplinary teams shared a mental model about the importance of team collaboration, communication with clients, and CS team members' programming skills. However, their perceptions about the completed apps were more divergent. CS students were more concerned about the correct functionality of the apps, while the HU students were much more concerned about the appearance of the app, usability and documentation of the app than the CS students. This mismatch in concerns is evidence of a mismatch of mental models between the HU and CS students, which we believe is mostly due to the poor communication and coordination within the teams. The three lectures given by a single instructor were also not sufficient to build a common mental model of the essential aspects of the apps.

Although both the HU and CS students thought that team collaboration was the most influential aspect of development, both HU and CS students complained about communication with team members from the other discipline. HU students' complaints were more general, while the CS students' complaints were more specific and directed at ineffective communication resulting in app failures. Many CS students expressed in the survey that they had insufficient time to complete the app. This could have created stress on the CS students and probably did not give them much time for communicating with HU students. Therefore, the time constraint of CS students could be one cause for the CS students' poor communication.

In addition, although the instructors attempted to enhance team coordination with documents that defined roles and commitments of team members, some teams still expressed the need for effective intra-team leadership. The instructors' documents did specify the role of a team leader. HU students mentioned that personality issues between HU and CS students caused communication barriers. Driskell et al. (2006) indicated that if personality traits fit into appropriate teamwork requirements, team effectiveness can be enhanced. We believe that the teams needed structure that included a leadership position assigned to both an HU and a CS student.

We also learned from the descriptive responses that CS team members attributed the poor collaboration to the difficulty in implementing the app's help documentation from their HU team members. However, HU students were concerned about the app's documentation and wanted the documentation implemented. During interviews, the HU students complained that CS students didn't see documentation as important. But the HU students did not evaluate their professional skills as influential to the app development as the CS students' professional skills. This circumstance of team members feeling that their efforts are not valuable can result in "social loafing." Karau & William (1993) highlighted that team members are "more likely to engage in social loafing when their individual outputs cannot be evaluated collectively? or their tasks are perceived "as low in meaningfulness or personal involvement." Consequently, many of the teams failed to implement their help documentation into the apps. We believe the cause of the lack of app documentation is a lack of coordination in the course assignments between the classes. Although the instructors attempted to coordinate assignments, the HU help documentation assignment did not have a definite due date and did not hold the HU students responsible for insuring the integration of the help document into the app.

To sum up the modifications in 2014, the instructors of the courses attempted to improve the collaboration by increasing the opportunities for communication, coordinating the assignments, defining stakeholders' roles and commitments, and sharing three lectures. These modifications were not sufficient to result in equal ratings of communications with the other discipline as within the same discipline. In addition the descriptive responses of 2014 interviews and surveys also indicated conflicts, mostly due to ambiguity of roles, lack of shared goals, lack of awareness of the values of the work from the other discipline, and the time constraint for implementing the app. The failures of the modifications are due to the implementation details of the course modifications. Additional opportunities for the teams to meet were not enough to ensure effective communication. Coordinating the assignments lacked specific due dates and did not hold team members responsible to ensure implementation. Documents defining stakeholder roles and commitments were not a sufficient substitute for team leaders. Three combined lectures given by one of the instructors were not enough to create a shared mental model of all the aspects of a successful app.

The insights from the survey results in 2014 have resulted in additional refinements to the development process and the collaboration between the 2015 courses. The instructors met frequently before the semester to integrate more assignments with definite due dates. The classes have had more joint lectures given by both instructors. Because of more meetings between the two course instructors and more efforts put on class coordination, the two instructors in 2015 shared more common goals than in previous years. At the beginning of the semester, two teamwork lectures were given in order to increase the awareness of the importance of the team collaboration and the contributions made by other disciplines. Teams were formed with two leaders, an HU student as product owner and a CS student as technical lead. In addition, teams developed mobile web apps instead of Android apps in order to take advantage of the scaffolding offered by web frameworks and to reduce the workload of CS students during the programming of the app.

Finally, the 2015 communication performance ratings showed promising results (see Table 5.6). The communication with the other discipline increased from previous years. The ANOVA results show that the CS students' ratings on HU team members increased significantly from 2013 to 2015 (see Table 5.9). The same increase occurs with the HU students' ratings on their CS team members (see Table 5.11). The post hoc Turkey HSD tests indicated both significant differences came from the increase in 2015 (see Table 5.10 and Table 5.10). Moreover, the 2015 communication ratings with the other discipline did not differ from the ratings with the same discipline (CS

Table 5.9. ANOVA Results for Communication Ratings Across Years (CS⇒HU)

	df	sum of sq	mean of sq	F value	*p* value
Between years	2	33.48	16.74	6.91	0.0019*
Residuals	65	157.43	2.42		

Table 5.10. Turkey HSD Post Hoc Testing on Variance of Communication Ratings (CS⇒HU)

	diff*	lwr*	upr*	*p* value
2013-2014	0.26	-0.87	1.39	0.846
2013-2015	1.62	0.45	2.80	0.004*
2014-2015	1.36	0.30	2.42	0.008*

*diff indicates the difference between means; lwr and upr are the lower and upper bonds of the 95% confidence interval.

Table 5.11. ANOVA Results for Communication Ratings Across Years (HU⇒CS)

	df	sum of sq	mean of sq	F value	*p* value
Between years	2	20.47	10.23	3.98	0.032*
Residuals	24	61.60	2.57		

with HU students: t (43) = -1.58, p = 0.12; HU with CS students: t (11) = 0, p = 1).

Table 5.12. Turkey HSD Post Hoc Testing on Variance of Communication Ratings (HU⇒CS)

	diff*	lwr*	upr*	p value
2013-2014	0.35	-1.55	2.25	0.890
2013-2015	2.02	0.07	3.96	0.040*
2014-2015	1.67	-0.17	3.50	0.080

*diff indicates the difference between means; lwr and upr are the lower and upper bonds of the 95% confidence interval.

5.2.4 Summary of the Interdisciplinary Collaboration

As software companies rely more on interdisciplinary teams for developing usable products, understanding the process of constructing a shared mental model becomes more essential. Although interdisciplinary collaboration can increase productivity, interdisciplinary collaboration has brought many problems because people from different disciplines talk in different "languages". We learned that interdisciplinary teams, especially novice teams, should have sufficient time, leadership and coordination to build this shared mental model for their product. Time constraints during development can result in software engineers neglecting communication with team members from other disciplines. Software engineers will choose to emphasize functionality at the cost of usability. Because software engineers control the implementation of the product, the final product can be lacking in usability and documentation unless the teams have a shared mental model and common goals. It is also necessary that team members are aware of the contribution by their teammates from other disciplines.

More meetings and coordinations before semester have led the two course instructors to shared more common goals on the project. More shared lectures were given by both instructors, which has helped to developed a shared language between the team members from two disciplines. Apart from these, the instructors gave assignments with firm deadlines, which led teams be clearer about their tasks and schedules. The instructors gave teams sufficient time to meet by giving them twenty minutes for discussion at the end of each class. The lectures about teamwork at the beginning of the semester have increased the students awareness of their team members' contribution and willingness to collaborate with the members from the other discipline. We believe that these course modifications have contributed to making the teams more collaborative. We also suggested the teams to have two leaders, performed by team members from both disciplines. However, not all of the teams had an effective leadership in 2015. Only in the Fisheye team, the two leaders coordinated the entire team, and they also communicated with each other, which has facilitated their team collaboration and performance.

5.3 Conclusion of Intra-Team Studies

We employed several techniques studying the software development teams. Focusing connection patterns between team members in the social network analysis illuminated the working structures of the six teams in 2015. The dynamic team process survey investigated the team members' perceptions on app satisfaction, working enjoyment, conflict, CS and HU team members' performances. We found that, the Deer team, a moderately centralized team, with team members from both disciplines actively involved in the team process, had better ratings than the Fisheye team, a strongly centralized team, and the Field team, a fully connected team. Although we learned from the descriptive responses that there were other issues in the Fisheye and Field teams (changing leadership and changing requirements), the outcome of the Deer team strongly suggests that team performance can be enhanced and facilitated by a moderate centralized, actively participating team, neither being very centralized with one or two strong person participation, nor being fully connected without effective leadership.

Tröster et al. (2014) argues the studies of teams should not focus only on a team's network structure, but also study how the team structure and demographic characteristics of team members interactively shape the team's outcome. We studied the collaboration with a focus on the disciplinary background of team members. When people from different disciplines work together, they are likely to consider what they are working on as more important and ignore what others are doing. For example, the CS team members would consider functionality the most essential aspect of an app, while the HU team members would consider usability testing important. Therefore, sufficient time, effective leadership and coordination are especially necessary for an interdisciplinary, creative team. We also surveyed the team members' characteristics using Gosling's personality survey (Gosling et al., 2003). However, we neither found any significant personality difference between the CS and HU team members, nor difference between groups.

Chapter 6

Conclusion

This study yielded a number of important findings. Table 6.1 summaries the all the findings of our studies and their insights to industrials. [1]

Prior to the study, we knew that communication is the key to a software development team's collaboration; thus, we assumed that more communication could lead to even better team performance. However, the study on the Lichen and Stream teams in 2013 indicated that the opposite was true. Whereas the Lichen team communicated with their client much more than did the Stream team, the Lichen team did not generate the levels of team performance we expected, particularly in the area of team efficiency; the stream team was better at implementing their scientist's app

[1] Part of the contents presented in this chapter have been published in the journal of Human Factor and Ergonomics (Zhang & Pastel, 2014, 2015).

Table 6.1. Summary of the Findings of the Study

Study	Findings	Insights
Study I	• The document from clients is important to software development process.	• The clients can play a critical role in the development process. Instead of considering clients as outside stakeholders, we should involve them as much as possible during the development process.
Study II	• The supporting document of the app helps to reduce the development time to complete the task, but decreases the number of creative features of an app.	• Teams can ask clients for supporting documents to explain better requirements. However, whether to use such a document depends on the task characteristics.
Study III	• Proper leadership with effective participation of other members can facilitate the development process.	• Appropriate leadership is important. A leader's role is more than just managing the team, but facilitating the team process. Teams solve problems, not leaders.
Study IV	• Modifications to the course on increasing the shared mental between disciplines improved collaboration and communication within team.	• Building a shared mental model among team members is critical to enhance interdisciplinary team performance.

requirements. After a more detailed investigation using a number of ethnographic techniques, we learned that a document from the scientist, which the team could extract detailed requirements from, was essential in facilitating the Stream team's development efficiency. To test the supporting effect of the document, we implemented an experimental study. The results of the experiment indicated that there were considerable performance differences between teams with and without the supporting document. The development teams that had the document completed the app interface prototype task faster than the teams without the document. This further supports the theory that a document supporting requirement elicitation can improve team efficiency and performance. However, we were concerned about the possible constraints the document might impose on the number of creative design ideas generated by teams. We tested our concerns by measuring the diversity of design ideas and found that teams with the document generated fewer ideas than the teams without the document. The supporting document significantly decreased the task completion time, but restricted the variety of ideas elicited from team members for the app interface design. Therefore, whether we should provide or request supporting documents in software development projects depends on the goals for the final product. Supporting documents are needed if a project is time-sensitive, which is typical in technology industries. Delaying a product's release schedule may cause untenable consequences, such as increased budgets, loss of customers, or the inability to be first to market. However, if a product's quality is more important than adhering to a specific timeline, a supporting document may not be a good choice.

Aside from analyzing the collaboration between the development teams and their clients, we also studied the intra-team collaboration. We investigated the teams from the perspectives of working structures, team mental models, and demographics, with a particular focus on interdisciplinary collaboration. Project satisfaction, individual performance and conflicts were used as measures of team performance.

Cognitive social structure surveys and social network analyses were used to capture the working structures of the 2015 teams. We observed different types of team structures: one very centralized team, two fully connected teams, and three moderately centralized teams. Team effectiveness was measured using the dynamic team process survey, which collected information about team members' mental models, including ratings on project satisfaction, working enjoyment, teammates' performance and conflicts. The dynamic team process survey also included descriptions of project goals and team conflicts. The responses were analyzed and compared from three perspectives: a comparison among six teams, a comparison of information gathered at different times of survey, and a comparison of information from members from two different disciplines.

Tröster et al. (2014) promoted a densely connected network, suggesting a positive relationship between the density of a network and a team's potency, particularly in a culturally diverse team. Later he found that centralization of the team network would be more likely to determine team performance. But how centralized should a team be? One of the moderately centralized teams, the Deer team, had obviously higher ratings on all of the questions compared to the very centralized team, the Fisheye team, and one connected team, the Field team. A moderately centralized team is more likely to achieve optimal performance: too little centralization leads to shortfalls and inefficiencies in the flow of information, and too dense of a network can cause "an overburdening of central individuals" in the team (Tröster et al., 2014). Apart from the degree of centralization, we also noticed that in the Deer team members from both disciplines participated and mingled better than did members on the other teams.

The responses from the ratings did not show a significant trend in their change over time. However, the descriptive responses indicated that some team members changed their goals from developing a good app to simply completing the app. More people aimed at meeting clients' requirements in the last survey than in the first and second survey. This may indicate a caveat brought by the study's context: a course project by student developers. Although Höst et al. (2000) suggested that using student developers to study software development can cause minor differences compared to using professional software developers in industry. The responses to project goals changed over time, which indicated that the students' motivations (completing the course and getting a good grade) on the project may be different from the motivations of professional developers in industry (making a good app and achieving career and salary goals). This encourages us to further the study in a real software industrial setting.

Between the years 2013 and 2015, several modifications to the course improved collaboration and communication within the team. These course modifications helped to establish the roles of all stakeholders involved in the project, make clear the commitment of individual team members to the project, provide a shared language for the team and clear commitment from the instructors to the multidisciplinary team effort. In addition, the course modifications strengthened the structure required for any team, meaning a place for teams to meet, sufficient time for meetings, a reasonable project outcome and sufficient time to achieve project outcomes. The communication ratings of other discipline on the team rose in the year 2015, so that there was no difference between the communication rating within the disciplines and with the other discipline.

Our results also showed that CS and HU team members reached a shared mental model in general, but with slight divergences. In the 2014 survey, we investigated what the students thought was influential to the project and what they were satisfied and dissatisfied with on the completed app. We learned that members from the two disciplines shared ideas about which factors were influential to the project: team collaboration, communication with clients and CS members' programming skills. However, they were concerned about different aspects of the app: CS members were most concerned about the correct functionality of the apps, while HU members were concerned about the appearance, usability and documentation of the apps. Comparing the responses to project goals between the two disciplines in 2015 yielded results similar to the 2014 survey: team members from the two disciplines had different focuses based on their discipline. CS team members aimed toward a functional app, while HU members aimed toward an easy and intuitive app. In the cognitive social structure survey, the CS and HU team members' slices were also compared. For each team, members were in agreement on the general working structure of the teams, but had slightly different perceptions of the HU members' connections with CS members. HU members considered themselves to be communicating with more team members than their CS team members thought. While it is unavoidable that people would think their own specialities more important, we learned from the study that this difference in mental models could impact team members' motivations on the project. How would motivation differences affect team performance? We do not know yet. A further study could investigate the influence of different motivations of team members from different disciplines on their teams' performances.

Studies on this industrial-standard software development contribute to a delineation of a number of factors that influence collaboration in software development; some of these factors lead to project success and some lead to failure. First, we learned the importance of stakeholders outside the development teams, such as clients. Better preparation with clients for the development teams could facilitate improved development schedule performance. Second, a team with neither a very centralized nor connected working structure, in which members from all disciplines actively participated, would be more likely to generate good performance. Software teams do need effective leadership. However, strong leadership may result in one or two people taking over all of the task implementation without others' participation. Moderately centralized software teams are most likely to shape satisfactory team performance. Finally, we need to build a shared mental model between team members from different disciplines. Many approaches could be employed in building this shared mental model. For example, a project manager can provide more opportunities for team members to meet. Some researchers believe that informal conversations are more crucial to teamwork because ideas are more likely to be shared and discussed (Pinton & Pinto, 1990). Informal conversations, such as caring greetings, personal conversations and story-telling, can establish a comfortable, supportive mood and encourage self-disclosure and sharing ideas with others, thus building stronger group trust (Holton, 2001). This is an issue of critical importance.

The understanding of this software development team process has provided us with many insights to improve the software development in industries. Effective leadership and appropriate team working structure can facilitate development process. Other than that, a software project does not only require a good communication channel and teamwork structure of development team itself, but also needs effective collaboration with stakeholders outside the development team. Stakeholders other than software developers, such as product client, can also play a critical role in determining project outcome. This may encourage a change of some current software teams' focus from the product itself to clients. The first step toward this change is to spend more time to prepare with clients before and during the development process. The interdisciplinary collaboration could be another challenge in software development. It is not realistic to require non-software engineers to study coding, instead project managers or administrators should create more opportunities for collaborators to communicate and increase their awareness of the importance and contribution of other disciplines. However, we also observed some limitations of the study, including the student developers' motivation changes across the semester from developing a good app to just completing it for grades. This could possibly created bias in studying team performance's influences and encourages us to further this study into an industrial setting.

References

Alexander, I. F. (2005). A Taxonomy of Stakeholders: Human Roles in System Development. *International Journal of Technology and Human Interaction*, *1*(1), 23-59.

Alexander, I. F., & Robertson, S. (2004). Understanding Project Sociology by Modeling Stakeholders. *Software, IEEE*, *21*(1), 23-27.

Ancona, D. G., & Caldwell, D. F. (1992). Demography and Design: Predictors of New Product Team Performance. *Organization Science*, *3*(3), 321-341.

Anthonisse, J. M. (1971). *The rush in a graph.* Amsterdam, the Netherlands: Mathematisch Centrum.

Avianca Flight 52. (1990). Retrieved from https://en.wikipedia.org/wiki/Avianca_Flight_52

Avnet, M. S. (2009). *Socio-cognitive Analysis of Engineering Systems Design:*

Shared Knowledge, Process, and Product (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Avnet, M. S., & Weigel, A. L. (2013). The Structural Approach to Shared Knowledge An Application to Engineering Design Teams. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *55*(3), 581-594.

Borgman, C. L., & Furner, J. (2002). Scholarly Communication and Bibliometrics. *Annual Review of Information Science and Technology*, *36*(1), 2-72.

Brands, R. A. (2013). Cognitive Social Structures in Social Network Research: A Review. *Journal of Organizational Behavior*, *34*(S1), S82-S103.

Cannon-Bowers, J. A., Salas, E., & Converse, S. A. (1993). Shared Mental Models in Expert Team Decision Making. *Current Issues in Individual and Group Decision Making*, 221-246.

Cashman, S. B., Reidy, P., Cody, K., & Lemay, C. A. (2004). Developing and Measuring Progress Toward Collaborative, Integrated, Interdisciplinary Health Care Teams. *Journal of Interprofessional Care*, *18*(2), 183-196.

Chatzoglou, P. D., & Macaulay, L. A. (1996). Requirements Capture and Analysis: A Survey of Current Practice. *Requirement Engineering*, *1*(2), 75-87.

Cockburn, A. (2006). *Agile software development: the cooperative game.* New York: Pearson Education.

Conboy, K., Coyle, S., Wang, X., & Pikkarainen, M. (2011). People over Process: Key Challenges in Agile Development. *IEEE Software*, *28*(4), 48-57.

Cooke, N. J., Gorman, J. C., & Rowe, L. J. (2008). An Ecological Perspective on Team Cognition. *Team Effectiveness in Complex Organizations: Cross-disciplinary Perspectives and Approaches*, 157-182.

Cooke, N. J., Salas, E., Cannon-Bowers, J. A., & Stout, R. J. (2000). Measuring Team Knowledge. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *42*(1), 151-173.

Cooke, N. J., Salas, E., Kiekel, P. A., & Bell, B. (2004). Advances in Measuring Team Cognition. *Team Cognition: Understanding the Factors that Drive Process and Performance*, 83-106.

Cronbach, L. J. (1957). The Two DIsciplines of Scientific Psychology. *American Psychologist*, *12*(11), 671-684.

Cross, N., & Cross, C. A. (1995). Observations of Teamwork and Social Processes in Design. *Design Studies*, *16*(2), 143-170.

Curtis, B., Soloway, E. M., Brooks, R., Black, J. B., Ehrlich, K., & Ramsey, H. R. (1986). Software Psychology: The Need for an Interdisciplinary Program. In *Proceedings of the IEEE* (Vol. 74, p. 1092-1106).

Cushman, J. H. (1990). *Avianca Flight 52: The Delays that Ended In Disaster.* Retrieved from http://www.nytimes.com/1990/02/05/nyregion/avianca-flight-52-the-delays-that-ended-in-disaster.html?pagewanted=all

Damian, D. (2007). Stakeholders in Global Requirements Engineering: Lessons Learned from Practice. *Software, IEEE*, *24*(2), 21-27.

Driskell, J. E., Goodwin, G. F., Salas, E., & O'Shea, P. G. (2006). What Makes a Good Team Player? Personality and Team Effectiveness. *Group Dynamics: Theory, Research, and Practice*, *10*(4), 249-271.

Dyer, J. L. (1984). Team Research and Team Training: A State of the Art Review. *Human Factors Review*, 285-323.

Easterbrook, J. A. (1959). The Effect of Emotion on Cue Utilization and the Organization of Behavior. *Psychological Review*, *66*(3), 183-201.

Edwards, B. D., Day, E. A., Arther, J. W., & Bell, S. T. (2006). Relationships among Team Ability Composition, Team Mental Models, and Team Performance. *Journal of Applied Psychology*, *91*(3), 727-736.

Espinosa, J. A., & Clark, M. A. (2014). Team Knowledge Representation: A Network Perspective. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *56*(2), 333-348.

Essentials of Educational Psychology. (2009). New York, NY: Pearson.

Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science*, *46*(12), 1554-1568.

Freeman, L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, *40*(1), 35-41.

Freeman, L. C. (1979). Centrality in Social Networks: Conceptual Clarification. *Social Networks*, *1*, 215-239.

Freeman, R. B., & McVea, J. (2001). *A Stakeholder Approach to Strategic Management* (Tech. Rep.). University of Virginia.

Freeman, R. B., & Medoff, J. L. (1984). *What Do Unions Do.* New York, NY: Basic Books.

Galegher, J., & Kraut, R. E. (1994). Computer-mediated Communication for Intellectual Teamwork: An Experiment in Group Writing. *Information System Research*, *5*(2), 110-138.

Gemuenden, H. G., & Lechler, T. (1997). Success Factors of Project Management: The Critical Few-an Empirical Investigation. In *Proceedings of innovation in technology management-the key to global leadership (PICMET'97): Portland international conference on management and technology* (p. 375-377).

Gilson, L. L., & Shalley, C. E. (2004). A Little Creativity Goes A Long Way: An Examination of Team's Engagement in Creative Process. *Journal of Management*, *30*(4), 453-470.

Goldschmidt, G. (1995). The Designer as A Team of One. *Design Studies*, *16*(2), 189-209.

Gonzales, A. L., Hancock, J. T., & Pennebaker, J. W. (2010). Language Style Matching as A Predictor of Social Dynamics in Small Groups. *Communication Research*, *37*(1), 3-19.

Gosling, S. D., Rentfrow, P. J., & Swann, J. W. B. (2003). A Very Brief Measure of the Big-FIve Personality Domains. *Journal of Research in Personality*, *37*(6), 504-528.

Griffin, A., & Hauser, J. R. (1992). Patterns of Communication among Marketing, Engineering and Manufacturing: A Comparison between two New Product Development Teams. *Management Science*, *38*(3), 360-373.

Grimble, R., & Chan, M. K. (1995). Stakeholder Analysis for Natural Resource Management in Developing Countries. *Natural Resources Forum*, *19*(2), 113-124.

Harmon-Jones, E., Price, T. F., & Gable, P. A. (2012). The Influence of Affective States on Cognitive Broadening/Narrowing: Considering the Importance of Motivational Intensity. *Social and Personality Psychology*, *6*(4), 314-327.

Hedges, L., & Olkin, I. (1995). *Statistical methods for meta-analysis.* San Diego, CA: Academic Press.

Hein, A. M., Tziolas, A. C., & Osborne, R. (2011). Project Icarus: Stakeholder Scenarios for An Interstellar Exploration Program. *Journal of the British Interplanetary Society*, *64*, 224-233.

Hoegl, M., & Gemuenden, H. G. (2001). Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. *Organization Science*, *12*(4), 435-449.

Hoegl, M., Parboteeah, K. P., & Gemuenden, H. G. (2003). When Teamwork Really Matters: Task Innovativeness as A Moderator of the Teamwork-Performance Relationship in Software Development Projects. *Journal of Engineering and Technology Management*, *24*(4), 281-302.

Holton, J. A. (2001). Building Trust and Collaboration in a Virtual Team. *Team Performance Management: An International Journal*, *7*(3-4), 36-47.

Höst, M., Regnell, B., & Wohlin, C. (2000). Using Students as Subjects-A Comparative Study of Students and Professionals in Lead-time Impact Assessment. *Empirical Software Engineering*, *5*(3), 201-214.

Jackson, D. W. (2010). Collaboration Versus Communication: Selecting the Appropriate Tool. *Law Libr J.*, *102*(2), 315-324.

Jobidon, M. E., Muller-Gass, A., Duncan, M., & Blais, A. R. (2012). The Enhancement of Mental Models and its Impact on Teamwork. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 56, p. 1703-1707).

Jones, C. (1996). *Applied software measurement.* New York, NY: McGraw-Hill.

Karau, S. J., & William, K. D. (1993). Social Loafing: A Meta-analytic Review and Theoretical Integration. *Journal of Personality and Social Psychology*, *65*(4), 681-706.

Keil, M., & Carmel, E. (1995). Customer-developer Links in Software Development. *Communication of the ACM*, *38*(5), 33-44.

Klein, G. A. (1998). *Sources of power: How people make decisions.* Cambridge, MA: MIT Press.

Klimoski, R., & Mohammed, S. (1994). Team Mental Model: Construct or Metaphor. *Journal of management*, *20*(2), 403-437.

Krackhardt, D. (1987). Cognitive Social Structures. *Social Networks*, *9*, 109-134.

Kraut, R. E., & Streeter, L. A. (1995). Coordination in Software Development. *Communication of the ACM*, *38*(3), 69-81.

Langan-Fox, J., Anglim, J., & Wilson, J. R. (2004). Mental Models, Team Mental Models, and Performance: Process, Development, and Future Directions. *Human Factors and Ergonomics in Manufacturing*, *14*(4), 331-352.

Leeders, R. T. A., Engelen, J. M. V., & Kratzer, J. (2003). Virtuality, Communication, and New Product Team Creativity: A Social Network Perspective. *Journal of Engineering and Technology Management*, *20*(1-2), 69-92.

Leon, M. (2004). *Do Developers have Poor Social Skills.* Retrieved from http://discuss.fogcreek.com/askjoel/default.asp?cmd=show&ixPost=5459

Lim, B. C., & Klein, K. J. (2006). Team Mental Models and Team Performance: A Field Study of the Effects of Team Mental Model Similarity and Accuracy. *Journal of Organizational Behavior*, *27*(4), 403-418.

Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., & Cannon-Bowers, J. A. (2000). The influence of shared mental models on team process and performance. *Journal of Applied Psychology*, *85*(2), 273-283.

Mathis, L. (2009). *Designers are not programmers.* Retrieved from http://ignorethecode.net/blog/2009/03/10/designers-are-not-programmers/

Mayer, A., Robert, R., Wallace, C., Oppliger, S., & Donovan, R. (2013). *Environmental Cybercitizens: Engaging Citizen Scientists in Global Environmental Change Through Crowdsensing and Visulization.* Retrieved from http://www.nsf.gov/awardsearch/showAward?AWD_ID=1135523

McIntyre, H. H., & Foti, R. J. (2013). The Impact of Shared Leadership on Teamwork Mental Models and Performance in Self-directed Teams. *Group Process & Intergroup Relations*, *16*(1), 46-57.

Moe, N. B., Dingsyr, T., & Dyb, T. (2010). A Teamwork Model for Understanding an Agile Team: A Case Study of a Scrum Project. *Information and Software Technology*, *52*(5), 480-491.

Mohammed, S., Ferzandi, L., & Hamilton, K. (2010). Metaphor No More: A 15-year Review of the Team Mental Model Construct. *Journal of Management*, *36*(4), 876-910.

Mueller, S. T., & Elizabeth, S. (2008). Cultural Mixture Modeling: A Method for Identifying Cultural Consensus. *ARA Technology Review*, *4*, 38-45.

Newman, M. E. (2003). The Structure and Function of Complex Networks. *SIAM review*, *45*(2), 167-256.

O'Hara, F. M. (2001). A Brief History of Technical Communication. In *Proceedings of the annual conference-society for technical communication* (Vol. 48, p. 500-504).

Olson, G. M., Olson, J. S., Carter, M. R., & Storrosten, M. (1992). Small Group Design Meetings: An Analysis of Collaboration. *Human Computer Interaction*, *7*(4), 347-374.

Orchard, C. A., Curran, V., & Kabene, S. (2009). Creating a Culture for Interdisciplinary Collaborative Professional Practice. *Medical Education Online*, *10*(11), 1-13.

Paetsch, F., berlein, A. E., & Maurer, F. (2003). Requirements Engineering and Agile Software Development. In *Proceedings of the twelfth IEEE international workshops on enabling technologies: Infrastructure for collaborative enterprises (WET ICE 2003)* (p. 308-313).

Pastel, R., Siegel, M., Zhang, W., & Mayer, A. (2015). Team Building in Multidisciplinary Client-sponsored Project Courses. *ACM Transactions on Computing Education (TOCE), Special Issue on Team Projects in Computing Education*, *15*(4), 19.

Phuwanartnurak, A. J. (2009). Interdisciplinary Collaboration through Wikis in Software Development. In *Proceedings of the ICSE workshop on wikis for software engineering (WIKIS4SE 2009)* (p. 82-90).

Pinton, M. B., & Pinto, J. K. (1990). Project Team Communication and Cross-functional Cooperation in New Program Development. *Product Innovation Management*, *7*(3), 200-212.

Pouloudi, A. (1999). Aspects of the Stakeholder Concept and Their Implications for Information Systems Development. In *Proceedings of the 32nd annual hawaii international conference IEEE.*

Qin, J., Lancaster, F. W., & Allen, B. (1997). Types and Levels of Collaboration in Interdisciplinary Research in the Sciences. *Journal of the American Society for information Science*, *48*(10), 893-916.

Roberts, N. C., & Bradley, R. T. (1991). Stakeholder Collaboration and Innovation: A Study of Public Policy Initiation at the State Level. *The Journal of Applied Behavioral Science*, *27*(2), 209-227.

Saeki, M. (1995). Communication, Collaboration and Cooperation in Software Development - How Should We Support Group Work in Software Development? In *Proceedings of 1995 asia pacific IEEE.*

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On Teams, Teamwork, and Team Performance: Discoveries and Developments. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(3), 540-547.

Shaw, M. E. (1954). Group Structure and the Behavior of Individuals in Small Groups. *Journal of Psychology*, *38*(1), 139-149.

Sommerville, I., & Sawyer, P. (1997). *Requirements enginneering - a good practice guide.* New York, NY: Jon Wiley & Sons.

Taylor, A., & Greve, H. R. (2006). Superman or the Fantastic Four? Knowledge Combination and Experience in Innovative Teams. *Academy of Management Journal*, *49*(4), 723-740.

Tröster, C., Mehre, A., & Knippenberg, D. V. (2014). Structuring for Team Success: The Interactive Effects of Network Structure and Cultural Diversity on Team Potency and Performance. *Organizational Behavior and Human Decision Processes*, *124*(2), 245-255.

Uitdewilligen, S., Waller, M. J., & Pitarju, A. H. (2013). Mental Model Updating and Team Adaptation. *Small Group Research*, *44*(2), 127-158.

Van de Ven, A. H., Delbecq, A. L., & Koenig, J. R. (1976). Determinants of Coordination Modes within Organizations. *American Sociological Review*, *41*(2), 322-338.

WIldman, J. L., Salas, E., & Scott, C. P. (2014). Measuring Cognition in Teams A Cross-Domain Review. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *56*(5), 911-941.

WIldman, J. L., Thayer, A. L., Pavlas, D., Salas, E., Stewart, J. E., & Howse, W. R. (2012). Team Knowledge Research: Emerging Trends and Critical Needs. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *54*(1), 84-111.

Wildman, J. L., Thayer, A. L., Pavlas, D., Salas, E., Stewart, J. E., & Howse, W. R. (2012). Team Knowledge Research: Emerging Trends and Critical Needs. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *54*(1), 84-111.

Yen, J., Yin, J., Joerger, T. R., Miller, M. S., Xu, D., & Volz, R. A. (2001). Cast: Collaborative Agents for Simulating Teamwork. In *Proceedings of the 17th international joint conference on artificial intelligence (IJCAI'2001)* (p. 1135-1144).

Yong, K., Sauer, S. J., & Mannix, E. A. (2014). Conflict and Creativity in Interdisciplinary Teams. *Small Group Research*, *45*(3), 266-289.

Zhang, W., & Pastel, R. (2014). Communication is More than Verbal: The Role of Clients' Documents in Requirement Solicitation. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 58, p. 1486-1490).

Zhang, W., & Pastel, R. (2015). Interdisciplinary Team Collaboration between Software Engineers and Technical Communicators. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 59, p. 1137-1141).

Appendix A

2013 Semester Survey

CS4760 Stakeholders Communication Survey

The purpose of this survey is to learn how you communicated with different stakeholders in your project. There are 4 sections of the survey that ask questions about the communication with different stakeholders (your teammates, HU undergraduate team, scientist and CS graduate student) and a final section about the course in general.

Your username (**wzhang5@mtu.edu**) will be recorded when you submit this form. Not **wzhang5**? Sign out
* Required

Section 1: Communication within your group (CS students only)

1. **Rate the overall quality of the communication within your team (CS team only).** *
Mark only one oval.

	1	2	3	4	5	6	
Very poor	◯	◯	◯	◯	◯	◯	Very good

2. **Why did you give the above rating for the quality of communication with your teammates?** *

3. **What challenges did you face in communicating with your CS teammates, and how did you address those challenges?** *

Communication Frequency within your Group

In the following 6 questions, please check the approximate usage frequency for each medium that you used to communicate with your team members. These communication media include face-to-face, Skype, phone, email, texting and an alternative communication medium.

4. **How frequently did you communicate IN-PERSON or face-to-face?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

5. **How frequently did you use SKYPE or another audiovisual medium to communicate with your teammates?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

6. **How frequently did you use the PHONE to communicate with your teammates?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

7. **How frequently did you use EMAIL to communicate with your teammates?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

8. **How frequently did you use TEXTING to communicate with your teammates?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

9. **How frequently did you use ALTERNATIVE communication medium with your teammates?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

10. **If you used an alternative communication medium, what was it?**

11. **Of these media, which was most effective?** *
select only one:
Mark only one oval.

- in-person
- Skype
- phone
- email
- texting
- alternative media

12. **Why was this medium the most effective for communicating with your teammates?** *

13. **What are some suggestions that you would give to CS students to improve the collaboration on this project?** *

Section 2: Communication with the Humanities (HU) undergraduate team

14. **Rate the overall quality of the communication between your team and the HU team.** *
Mark only one oval.

	1	2	3	4	5	6	
Very poor	◯	◯	◯	◯	◯	◯	Very good

15. **Rate the overall quality of the communication between your team and the HU team.** *
Mark only one oval.

	1	2	3	4	5	6	
Very poor	◯	◯	◯	◯	◯	◯	Very good

16. **Why did you give this rating for the communication with the CS undergraduate team?** *

17. **What challenges did you face in communicating with your HU teammates, and how did you address those challenges?** *

Communication Frequency with HU Group

In the following 6 questions, please check the approximate usage frequency for each medium that you used to communicate with your team members. These communication media include face-to-face, Skype, phone, email, texting and an alternative communication medium.

18. **How frequently did you communicate IN-PERSON or face-to-face with the HU team?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

19. **How frequently did you you use SKYPE or another audiovisual medium to communicate with the HU team?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

20. **How frequently did you you use the PHONE to communicate with the HU team?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

21. **How frequently did you you use the EMAIL to communicate with the HU team?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

used to communicate with your team members. These communication media include face-to-face, Skype, phone, email, texting and an alternative communication medium.

18. **How frequently did you communicate IN-PERSON or face-to-face with the HU team? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

19. **How frequently did you you use SKYPE or another audiovisual medium to communicate with the HU team? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

20. **How frequently did you you use the PHONE to communicate with the HU team? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

21. **How frequently did you you use the EMAIL to communicate with the HU team? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

22. **How frequently did you you use the TEXTING to communicate with the HU team?** *

Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

23. **How frequently did you you use an ALTERNATIVE communication medium to communicate with the HU team?** *

Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

24. **If you used an alternative communication medium, what was it?**

25. **Of these media, which was most effective medium for communicating with the HU team?**

Mark only one oval.

- in-person
- Skype
- phone
- email
- texting
- alternative communication medium

26. **Why was this medium the most effective for communicating with the HU team?** *

27. **Rate the extent to which you think your HU teammates' contributions improved the usability of the android app.** *
Mark only one oval.

	1	2	3	4	5	6	
didn't improve it at all	⬭	⬭	⬭	⬭	⬭	⬭	improved it significantly

28. **Why did you give you give the HU team's contribution this rating?** *

29. **What are some suggestions that you would give to HU students to improve the collaboration on this project?**

Section 3: Communication with your Scientist

30. **Rate the overall quality of the communication between your team and the scientist associated with your project.** *
Mark only one oval.

	1	2	3	4	5	6	
very poor	⬭	⬭	⬭	⬭	⬭	⬭	very good

31. **What do you think of the frequency of communication with the scientist?** *
Mark only one oval.

	1	2	3	4	5	6	
not enough communication	⬭	⬭	⬭	⬭	⬭	⬭	too much communication

32. **Who proposed the meeting usually?** *
Select more than one answer if necessary:
Check all that apply.

- [ ] Scientist
- [ ] CS students
- [ ] Humanities students
- [ ] All

33. **What challenges did you face in communicating with the scientist, and how did you address those challenges?** *

Communication Frequency with your Scientist

In the following 6 questions, please check the approximate usage frequency for each medium that you used to communicate with your scientist. These communication media include face-to-face, Skype, phone, email, texting and an alternative communication medium.

34. **How frequently did you communicate IN-PERSON or face-to-face with your scientist?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- ( ) never
- ( ) once or twice a semester
- ( ) once or twice a month
- ( ) once or twice a week
- ( ) nearly daily

35. **How frequently did you use SKYPE or another audiovisual medium to communicate with your scientist?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- ( ) never
- ( ) once or twice a semester
- ( ) once or twice a month
- ( ) once or twice a week
- ( ) nearly daily

36. **How frequently did you use the PHONE to communicate with your scientist?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

37. **How frequently did you use EMAIL to communicate with your scientist?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

38. **How frequently did you use TEXTING to communicate with your scientist?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

39. **How frequently did you use an ALTERNATIVE communication medium not listed above to communicate with your scientist?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

40. **If you used an alternative communication medium, what was it?**

36. **How frequently did you use the PHONE to communicate with your scientist? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

37. **How frequently did you use EMAIL to communicate with your scientist? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

38. **How frequently did you use TEXTING to communicate with your scientist? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

39. **How frequently did you use an ALTERNATIVE communication medium not listed above to communicate with your scientist? ***
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

40. **If you used an alternative communication medium, what was it?**

41. **Of these media, which was most effective for communicating with your scientist?** *
Mark only one oval.

- ◯ in-person
- ◯ Skype
- ◯ phone
- ◯ email
- ◯ texting
- ◯ alternative communication

42. **Why was this the most effective medium for communicating with your scientist?** *

....

....

....

....

....

43. **Rate the extent to which you think the scientists' input and contributions improved the usability of the android app.** *
Mark only one oval.

	1	2	3	4	5	6	
didn't improve it at all	◯	◯	◯	◯	◯	◯	improved it significantly

44. **Why did you give this rating for your scientist's contribution to improving the usability of the app?** *

....

....

....

....

....

Section 4: Communication with the CS Graduate Student

45. **Rate the overall quality of the communication within between your team and the CS graduate student assigned to your project.** *
Mark only one oval.

	1	2	3	4	5	6	
very poor	◯	◯	◯	◯	◯	◯	very good

46. **Why did you give this rating for the quality of communication with the CS graduate student?** *

47. **What challenges did you face in communicating with the CS graduate student, and how did you address those challenges?** *

Communication Frequency with CS Graduate Student

In the following 6 questions, please check the approximate usage frequency for each medium that you used to communicate with the CS graduate student. These communication media include face-to-face, Skype, phone, email, texting and an alternative communication medium.

48. **How frequently did you communicate IN-PERSON or face-to-face with the CS graduate student?** *

Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

49. **How frequently did you use SKYPE or another audiovisual medium to communicate with the CS graduate student?** *

Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

50. **How frequently did you use the PHONE to communicate with the CS graduate student?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

51. **How frequently did you use EMAIL to communicate with the CS graduate student?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

52. **How frequently did you use TEXTING to communicate with the CS graduate student?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

53. **How frequently did you use an alternative communication media not listed above to communicate with the CS graduate student?** *
Make your best estimate of the average frequency and select only one:
Mark only one oval.

- never
- once or twice a semester
- once or twice a month
- once or twice a week
- nearly daily

54. **If you used an alternative communication medium, what was it?**

55. **Of these media, which was most effective medium for communicating with the CS graduate student?** *
Mark only one oval.

- ( ) in-person
- ( ) Skype
- ( ) phone
- ( ) email
- ( ) texting
- ( ) alternative communication

56. **Why was this the most effective medium for communicating between you and the CS graduate student?** *

57. **Rate the extent to which you think the CS graduate student's input and contributions improved the usability of the android app.** *
Mark only one oval.

	1	2	3	4	5	6	
didn't improve it at all	( )	( )	( )	( )	( )	( )	improved it significantly

58. **Why did you give this rating for the contribution of usability from the CS graduate student?** *

Section 5: General Course Questions

59. **What app did you work on?** *
Mark only one oval.

- ( ) Lichen
- ( ) Beach
- ( ) Tracking
- ( ) ROV
- ( ) Stream Features

60. **What did you learn from this course about communicating with different stakeholders?** *

61. **What ideas do you have for improving the course?** *

☐ Send me a copy of my responses.

Appendix B

2014 Semester Survey

CS4760 Course Survey 2014

Your username (**wzhang5@mtu.edu**) will be recorded when you submit this form. Not **wzhang5**?
Sign out
* Required

1. **What App did you work on? ***
Check all that apply.

- [ ] Watershed
- [ ] Tree walkers
- [ ] Water level
- [ ] Shipwreck
- [ ] Goggle-fox
- [ ] Mega Crystals

2. **Please rate your satisfaction to your completed App. ***
Mark only one oval.

	1	2	3	4	5	6	7	
Not satisfied	( )	( )	( )	( )	( )	( )	( )	Very satisfied

3. **What aspect of the completed app are you most satisfied with? ***
Check all that apply.

- [ ] A completed App
- [ ] Covering all the functionality request by the scientists
- [ ] Appearance of the App
- [ ] Correct functionality
- [ ] Other: ____________________

4. **What aspect of the completed app are you least satisfied with? ***
Check all that apply.

- [ ] A completed App
- [ ] Covering all the functionality request by the scientists
- [ ] Appearance of the App
- [ ] Correct functionality
- [ ] Other: ____________________

5. **What has influenced the development of your App?** *
You may choose more than one.
Mark only one oval.

- ( ) Collaboration within your team
- ( ) Communication with your scientist
- ( ) Documents from your scientist
- ( ) Cooperation with HU team members
- ( ) CS team members' skills (programming skills)
- ( ) HU team members' skills
- ( ) Team members's personality
- ( ) Time you had to implement your App
- ( ) Other: ____________

6. **What most influenced the development of your App?** *
Choose only one.
Check all that apply.

- [ ] Collaboration within your team
- [ ] Communication with your scientist
- [ ] Documents from your scientist
- [ ] Cooperation with HU team members
- [ ] CS team members' skills (programming skills)
- [ ] HU team members' skills
- [ ] Team members's personality
- [ ] Time you had to implement your App
- [ ] Other: ____________

7. **Rate how you enjoyed the development process working with your teammates.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	( )	( )	( )	( )	( )	( )	( )	Very much

8. **Rate how you enjoyed the development process working with your scientist.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	( )	( )	( )	( )	( )	( )	( )	Very much

9. **Were there any conflicts during the design or development of your App?** *
You may choose more than one.
Mark only one oval.

- ( ) Yes
- ( ) No
- ( ) Other: ______________

10. **Rate the severity of the worst conflict, if there was one.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Very Bad	( )	( )	( )	( )	( )	( )	( )	Not Very Bad

11. **What do you think was the cause of the most severe conflict, if there was a conflict?** *

____
____
____
____
____

12. **What do you think your team could have done to improve the development process?** *

____
____
____
____
____

13. **Was there a member in your group that took charge of the team or lead the team?** *
Check all that apply.

- [ ] Yes
- [ ] No
- [ ] Other: ______________

14. **Rate the average CS team members' performance.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad	( )	( )	( )	( )	( )	( )	( )	Very good

15. **Rate the average HU team members' performance.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Very bad	◯	◯	◯	◯	◯	◯	◯	Very good

16. **Rate the degree that you worked on the App only to complete the course.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Disagree	◯	◯	◯	◯	◯	◯	◯	Agree

17. **Rate the overall quality of the communication between your team and the HU team.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Very Poor	◯	◯	◯	◯	◯	◯	◯	Very Good

18. **Rate the overall quality of the communication between your team and the scientist associated with your project.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Very Poor	◯	◯	◯	◯	◯	◯	◯	Very Good

19. **Rate the overall quality of the communication within your team (CS team only).** *

Mark only one oval.

	1	2	3	4	5	6	7	
Very Poor	◯	◯	◯	◯	◯	◯	◯	Very Good

20. **How frequently did you communicate with the HU team?** *

Make your best estimate of the average frequency and select only one:
Check all that apply.

- [ ] Never
- [ ] Once during the semester
- [ ] Twice during the semester
- [ ] Once a month
- [ ] Twice a month
- [ ] Once a week
- [ ] More than once a week
- [ ] Nearly daily

21. **How frequently did you communicate with your scientist?** *

Make your best estimate of the average frequency and select only one:
Check all that apply.

- [ ] Never
- [ ] Once during the semester
- [ ] Twice during the semester
- [ ] Once a month
- [ ] Twice a month
- [ ] Once a week
- [ ] More than once a week
- [ ] Nearly daily

22. **How frequently did you communicate with your CS teammates?** *

Make your best estimate of the average frequency and select only one:
Check all that apply.

- [ ] Never
- [ ] Once during the semester
- [ ] Twice during the semester
- [ ] Once a month
- [ ] Twice a month
- [ ] Once a week
- [ ] More than once a week
- [ ] Nearly daily

23. **What have you learned from the course?** *
You may choose more than one.
Mark only one oval.

- ( ) Software development
- ( ) Communication with scientist (client)
- ( ) Collaboration with teammates
- ( ) Collaboration with people in other departments
- ( ) Other: ____________

24. **What ideas do you have for improving the course?** *

____
____
____
____
____

25. **What experience did you have working with your team members including humanities students?** *

____
____
____
____
____

- [ ] Send me a copy of my responses.

Powered by Google Forms

Appendix C

Scientsts-Student Agreement

Scientist – Student Agreement

Robert Pastel, 10/31/2013 10:28 AM

This document explains the relationship and the agreement between students in the Computer Science courses, Humanities course and Scientists sponsoring Android applications. It details the expectations of responsibilities, time and effort to be committed by the sponsoring scientists and students.

1. Development Process

1.1 Computer Science and Humanities Courses

Three courses are involved in the Android app applications:

1. CS4760 – Undergraduate Computer Science course in Human-Computer Interaction Design and Implementation.
2. CS5760 – Graduate Computer Science course in Human-Computer Interaction Evaluation and Testing
3. HU4628 – Undergraduate Humanities course in Usability and Instructional Writing

All three courses introduce user-centered design by involving student teams in projects developing citizen science apps for sponsoring scientists. Typically, the teams are composed of 4 to 5 students from CS4760, two to three students from HU4628 and one to two students from CS5760. The CS4760 students are primarily responsible for design and coding the app. The HU4628 students are primarily responsible for design and help documentation for the app. The HU4628 students also preform usability testing of the help documentation. The CS5760 students are primarily responsible for evaluation and usability testing of the apps.

The Android app development process includes:

1. Gather app requirements
2. Delineate potential app users
3. Determine tasks to perform on app
4. Determine help and assistance that user of the apps will need
5. Delineate usability issues
6. Initial paper prototype design
7. Refinement of design
8. Final design
9. Adjust final design

10. Implement app
11. Design usability test
12. Usability testin
13. Analyze usability test results

The students attempt to accomplish all these design, development and evaluation steps in one semester, January 13 to April 24, less than 4 months. The teams need the scientists to provide timely input during the entire process..The goal is to implement the apps with sufficient functionality and fidelity so that the end product can be fairly evaluated, and the students' products can serve as a reliable and stable platform for further development of the apps.

1.2 Expected Schedule

A spreadsheet of the CS4760 and CS 5760 schedule can be downloaded from the CS4760 course website:

http://cs4760.csl.mtu.edu/2014/

Below is an expect schedule for major design steps during the semester:

1/13/2014 – First day of class

1/24/2014 – First meeting between teams and scientists

1/27/2014 – Second follow-up meeting with teams and scientists

2/10/2014 – 2/14/2014 – Paper prototype design presentation

2/12/2014 – 2/19/2014 – Feedback from scientist on the paper prototype

3/31/2014 – 4/4/2014 – Final design presentation

4/2/2014 – 4/9/2014 – Feedback from the scientist for adjustment of the final design

4/21/2014 – 4/25/2014 – Usability test result presentations

4/23/2014 – 4/29/2014 – Feedback from scientists on usability test results

5/3/2104 – Last day of the semester includes finals week

2. Scientist Effort

The scientist's participation is required throughout the design, implementation and evaluations of the apps. In particular, the scientist is required to conceive an app idea, assist in requirement gathering and design, and provide feedback on the design and evaluations. Below describes some of the details of these requirements.

2.1 App Concept

The scientist is responsible for conceiving and communicating the app concept. The app concept is communicated in the 1/24/2014 meeting with teams and refined during the 1/27/2014 follow up meeting with teams. I will schedule these meetings with the scientists before the semester begins.

I have two suggestions when conceiving app ideas:

1. The app should be as simple as possible and clearly stated in one or two sentences. Complicated apps, apps with many options cannot be completed in single semester. Apps that cannot be concisely explained will not be understood by the teams and gathering all the requirements will not be difficult.

2. Before the semester, the scientist and I will meet so that I can understand your app idea and assure that it is appropriately defined for the courses. This meeting typically takes one hour.

I will need your contact information before the semester begins and a brief description of the app, so the teams will have the material to select apps and contact scientists. I will need a document with:

1. Scientist Name
2. Scientist Title (description of your scientific expertise.)
3. Email address
4. Phone number (Only to be used as backup for conferencing.)
5. Conferencing Number or ID (If the scientist is to use conference number, skype etc.)
6. Office address
7. Descriptive App Name (This is not necessarily the name that the students will give their app.)
8. Paragraph description of App

The paragraph description of the app is not a description/specification of the design; rather it should motivate the app and give the general idea of the app. I have attached a document with last year's app descriptions to use as examples.

For scientists not at Michigan Technology University, I will arrange the venue for conducting meetings with the teams. This can be a regular phone, conference phone, skype, google hangout or other software.

2.2 App Design

The scientist should assist with the app design. Components of the design that the scientists should consider are:

1. Tasks that the app will perform
2. Who will use the app
3. Where the app will be used and any special circumstances during use
4. What data the app will collect

5. How to save the data
6. How the data will be collected
7. How the data will be presented
8. Sequence of events using the app
9. Any graphical design of the app

Scientists do not have to have address all of the above components, but these are the types of issues that the teams will need to answer. The amount of participation in the specification of these components is up to the scientist, but the more the scientist participates, the more precisely the vision of the app can be coordinated and the more successful the project will be.

2.3 Meetings and Presentations with Students

Scientists will need to participate in the app design process by meeting with the team and watching team presentations. I have found that the best mode of communication is face to face, followed by conference calling or video conferencing using phone, skype or google hang out, and that email correspondence the least effective. Nevertheless, all modes of communication will need to be used, except in the case of off-campus scientists, where face-to-face communication is unlikely.

Following are scheduled meeting dates and the expected focus of each meeting.

> 1/24/2014 – First meeting between teams and scientists. The teams will learn about the app idea from the scientist; including what tasks are to be performed with the app, who will use the app, where the app is to be used, etc. The meeting should last between 30 – 60 minutes.
>
> 1/27/2014 – Second follow-up meeting with teams and scientists. This second meeting allows teams to ask follow up questions after they have thought about the app idea over the weekend. The meeting should last between 30 – 60 minutes.
>
> 2/10/2014 – 2/14/2014 – Paper prototype design presentation. In this presentation, teams present their designs working through the apps' tasks using a design of the app expressed as drawings on paper. Each team has 30 minutes to present their design, answer questions and received suggestions from the rest of the teams and scientists. The presentation will be scheduled and videotaped. Scientists only need to attend presentation for their app. It is best if the scientist attend the presentation live, but scientists can view the videos later. Links to the video will be posted on the course website the evening of the presentation or the following day.
>
> 2/12/2014 – 2/19/2014 – Feedback from scientist on the paper prototype. During this meeting, scientists can provide detailed feedback of the design presented in the paper prototype presentation. The meeting should last between 30 – 60 minutes.
>
> 3/31/2014 – 4/4/2014 – Final design presentation. In this presentation, teams present their design by working through apps' tasks using a partial implemented app. Each team has 30 minutes to present their design, answer question and received suggestions from the rest of the teams and scientists. The presentation will be scheduled and videotaped. Scientists only need to attend presentation of their app. It is best if the scientist can attend the presentation live, but scientists

can view the videos later. Links to the video will be posted on the course website the evening of the presentation or the following day.

4/2/2014 – 4/9/2014 – Feedback from the scientist for adjustment of the final design. During this meeting, scientists can provide detailed feedback on the presented design. Because the students should have already implemented a significant portion of the app, only minor adjustments can be made. The teams are general good judge if they can implemented the adjustment. The meeting should last between 30 – 60 minutes.

4/21/2014 – 4/25/2014 – Usability test result presentations. During the prior week, teams will have been testing their app, using students from other classes. In this presentation, graduate students and humanities students present preliminary results of the usability tests. Each team has 30 minutes to present their design, answer question and received suggestions from the rest of the teams and scientists. The presentation will be scheduled and videotaped. Scientists only need to attend presentation of their app. It is best if the scientist could attend the presentation live, but scientists can view the videos later. Links to the video will be posted on the course website the evening of the presentation or the following day.

4/23/2014 – 4/29/2014 – Feedback from scientists on usability test results. During this meeting, graduate students give details of the usability test results. Scientists, teams and graduate students can discuss changes to a new version of the app as a result of the usability test results. The meeting should last between 30 – 60 minutes.

In addition, there other two scheduled class meetings that scientist may want to attend. At the beginning of the semester, on 1/20/2014 at 6 pm, we have a "get acquainted" meeting, and at the end of the semester, 4/25/2014 (expected) at 6 pm, we plan a final reception meeting. These meetings are opportunities for all stakeholders in the project to get to know each other and celebrate socially. Attendance is not required, and we'll not be able to video the events, but it may be possible to attend by conference calling.

The above schedule implies that there are 9 "scheduled meetings" of about one half hour each, but additional meetings for gathering app requirements may also be needed in the first seven weeks of the semester, depending on the complexity of the app and how precisely the app concept has been defined prior to the beginning of the project. The teams will be responsible for arranging for the meetings. They will try to minimize any extra meetings, but it is important for the the scientists to respond in a timely manner to the teams' requests for further meetings.

Note that the teams maintain a website for posting their design documents. Links to the teams' project website will be available on the CS4760 course website. The project website will be continuously available after the second week of the semester. Scientists will find these websites invaluable for learning about app design details and progress.

3. Student Team Effort

The student teams are responsible for the design, implementation and evaluation of the apps. Teams are not responsible for the initial app idea, but they should ensure that they understand the app idea and gather all the requirements for implementing the app. Below are details of the student teams' responsibilities with regard to implementing the app.

3.1 App Design, Implementation and Evaluation

Students in CS4760 and HU4628 are jointly responsible for gathering requirements and the design of the app. Students from both classes are encouraged to contribute to delineating

1. App users
2. App tasks
3. App graphical design
4. App details, such as wording etc.
5. App usability issues

Students in CS4760 and HU4628 are jointly responsible for maintaining their projects' websites.

Students in CS4760 are responsible for the implementation/coding of the Android app, but students in HU4628 may and and are encouraged to contribute to the view layouts. View layouts are expressed in XML similar to HTML layouts.

Students in HU4628 are responsible for the "verbal content" of the app and specifically help documentation required by the app, but students in CS4760 may and can advise or edit the app content. In addition, HU4628 students will design and administrate a usability test of the app content.

More details about the responsibility of CS4760 and HU4628 students are available in the assignments for the courses.

Students in CS5760 are responsible for assisting in identifying users and making a hierarchal task analysis of the app. Most important, students in CS5760 are responsible for evaluating the app design during the semester and at the end of the semester preform a usability test. Students in CS4760 and HU4628 will assist with the usability testing.

Students in CS5760 are also responsible for maintaining their own website for posting their evaluations.

More details about the responsibility of CS5760 students are available in the assignments for the course.

3.2 Meetings and Presentations with Scientists

I will try to arrange for the first two meetings with scientists, but the teams are responsible for arranging the additional meetings. Three additional meeting are required after the paper prototype, final design presentation and usability test result presentations.. The meeting should occur as soon as possible after team's presentations:

> 2/12/2013 – 2/19/2013 –Feedback from scientist on the paper prototype. During this meeting between scientists and teams, scientists will provide detail feedback of the design presented in the paper prototype presentation. The meeting should last between 30 – 60 minutes.
>
> 4/2/2013 – 4/19/2013 – Feedback from the scientist for adjustment of the final design. During this meeting between scientists and teams, scientists will provide detailed feedback of the design presented in the final design presentation. Because the students should have already implemented

a significant portion of the app, only minor adjustments can be made. The teams are general good judge if they can implemented the adjustment. The meeting should last between 30 – 60 minutes.

4/23/2013 – 4/29/2013 – Feedback from scientists on usability test results. Grad students will describe the usability test results. Scientists, teams and graduate students can discuss changes to a new version of the app. The meeting should last between 30 – 60 minutes.

For all the "scheduled meetings," the teams are responsible for

1. Arranging the exact date and time for the "scheduled meeting" during the assigned week.
2. Planning for the meeting by typing pre-meeting notes
3. Recording meeting notes during the meetings
4. Posting on the project website the meeting notes.

Teams may need more meetings to assure that they completely understand the app and have gathered all the requirements. Teams should ensure that they gathered all the requirements early in the semester, but it is not unusual for a requirement to be discovered only late in the design. Try to minimize the number and impact these late requirements, by understanding as much of the app design early in the semester. By the fourth week of the semester the teams should have gathered all the important requirements. Teams may need additional meetings with the scientists to gather these requirements and will have the same responsibilities as those listed for the "scheduled meetings." Teams should be careful to minimize these meetings by being respectful of the scientists' time.

By the seventh week of the semester, the teams should have gathered all detailed requirements.

Appendix D

Supporting Document

Deer and Deer-Vehicle Collision Data Sheet

Observations (Example)	**Date**	**Location** (GPS or Landmark)	**Deer Type** (Buck, doe or fawn)	**Collision?**	**Number of deer**	**Time since Collision**	**User Name**
1	09/10/2015	47.119512, -88.564597	Doe	Yes	1	1 day	wzhang5
2	09/25/2015	46.394859, -85.382985.	Buck	No	1	NA	pastel

Figure D.1 An Example of Supporting Document. We made the supporting document based on the information provided by the Deer team scientist in 2015 for use of the experimental group.

Appendix E

User Story of the App

User story: Deer and Deer-Collision Reporter

According to the National Safety Council, there were 530,000 animal-related accidents in 2003 and these collisions resulted in 100 deaths and 10,000 injuries. The average cost per insurance claim, when you factor in auto claims involving bodily injury, is $10,000. The proposed app is to report the location of deer on the side of road and the deer-vehicle collisions. This will provide information about the migration patterns of deer populations and to locate the most dangerous sections of roadway for divers. The users of the application should be able to specify where and when they see the deer or deer collision, as well as the number and the type of deer.

This information can be used by transportation officials to consider mitigation strategies, such as deer crossing signs or roadside sensors that caution vehicles when deer are active. It could also be used to alert the drivers of conditions with elevated risk by looking up specific routes and reminding them to use high-beam headlights and reduce speed.

Appendix F

A Sample of the Prototyped App Interface

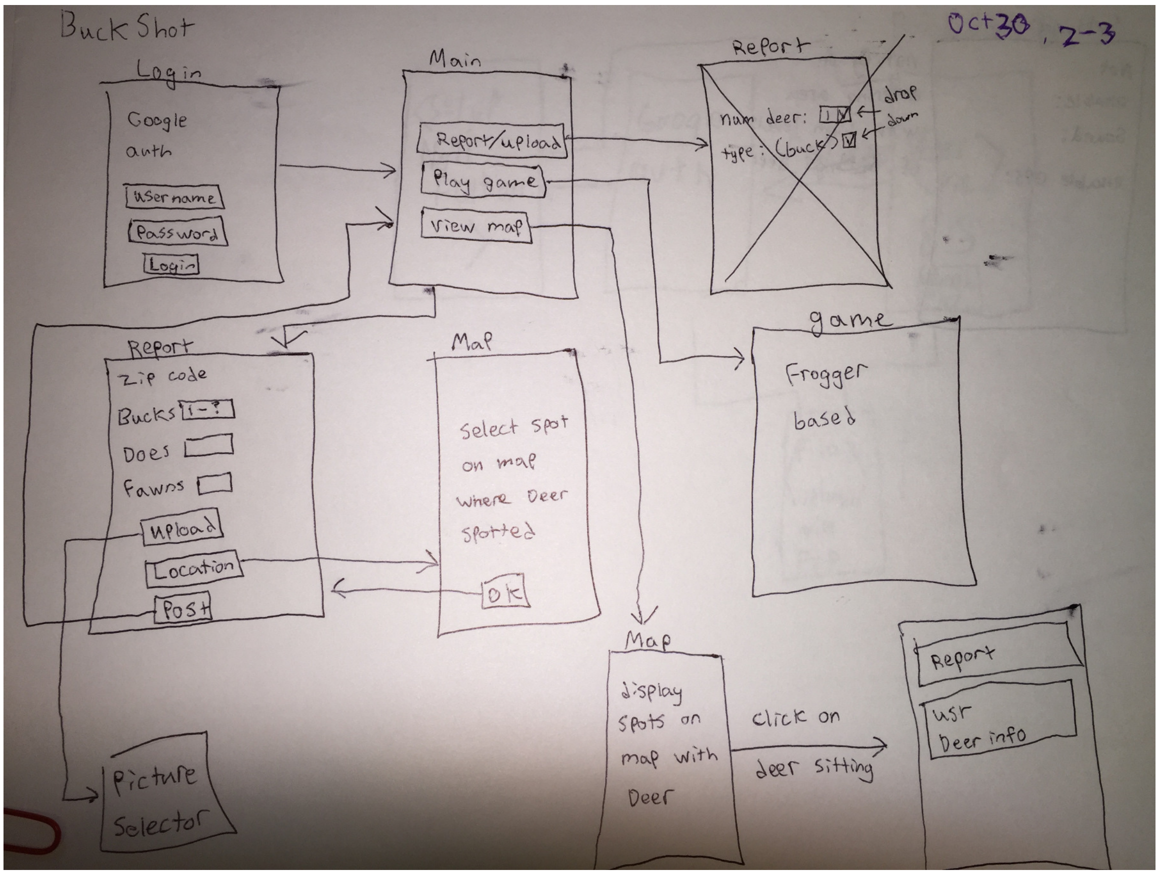

Figure F.1 A Sample of the Prototype of the App User Interface. This is the app interface prototype from the first group of participants in the experiment.

Appendix G

Post-study Survey

Post-study Survey

1. **What department are you from?**
   Mark only one oval.
   - ( ) Computer Science
   - ( ) Humanities
   - ( ) Other: ....

2. **What is your gender?**
   Mark only one oval.
   - ( ) Male
   - ( ) Female
   - ( ) Don't want to tell

3. **What class level are you in?**
   Mark only one oval.
   - ( ) Freshman
   - ( ) Sophomore
   - ( ) Junior
   - ( ) Senior
   - ( ) Graduate
   - ( ) Other: ....

4. **Have you worked on any project, including course project, that potentially be used by the public, these could be software, documents...?**
   Mark only one oval.
   - ( ) Yes
   - ( ) No

5. **Select the ideas that you have IMPLEMENTED in the final design of your app.**
Check all that apply.

- [ ] Location by map
- [ ] Location by GPS
- [ ] Location by place name (highway, nearest town)
- [ ] Listing deer or deer collision by date
- [ ] Listing deer or deer collision by highway
- [ ] Identify deer or deer collision
- [ ] Number of deer
- [ ] Type of deer
- [ ] Recording user name
- [ ] Time since collision if it is a deer collision
- [ ] Which side of the roadsides
- [ ] Suggesting routes to avoid deer collision
- [ ] Suggesting using high-beam headlights on high-risk routes during night
- [ ] Suggesting reducing speed while driving
- [ ] Rating route for high risk of hitting a deer
- [ ] Please list all other ideas that are not mentioned above:___________________________

6. **Select the ideas that you have CONSIDERED, VOCALIZED to your teammates, but NOT ADOPTED in the final design of your app.**
Check all that apply.

- [ ] Location by map
- [ ] Location by GPS
- [ ] Location by place name (highway, nearest town)
- [ ] Listing deer or deer collision by date
- [ ] Listing deer or deer collision by highway
- [ ] Identify deer or deer collision
- [ ] Number of deer
- [ ] Type of deer
- [ ] Recording user name
- [ ] Time since collision if it is a deer collision
- [ ] Which side of the roadsides
- [ ] Suggesting routes to avoid deer collision
- [ ] Suggesting using high-beam headlights on high-risk routes during night
- [ ] Suggesting reducing speed while driving
- [ ] Rating route for high risk of hitting a deer
- [ ] Please list all other ideas that are not mentioned above:___________________________

7. **Select the ideas that you have CONSIDERED, but DID NOT VOCALIZED to your teammates.**
Check all that apply.

- [ ] Location by map
- [ ] Location by GPS
- [ ] Location by place name (highway, nearest town)
- [ ] Listing deer or deer collision by date
- [ ] Listing deer or deer collision by highway
- [ ] Identify deer or deer collision
- [ ] Number of deer
- [ ] Type of deer
- [ ] Recording user name
- [ ] Time since collision if it is a deer collision
- [ ] Which side of the roadsides
- [ ] Suggesting routes to avoid deer collision
- [ ] Suggesting using high-beam headlights on high-risk routes during night
- [ ] Suggesting reducing speed while driving
- [ ] Rating route for high risk of hitting a deer
- [ ] Please list all other ideas that are not mentioned above:_________________________

8. **Rate the overall quality of the communication with your teammates.**
Mark only one oval.

	1	2	3	4	5	6	7	
Very poor	( )	( )	( )	( )	( )	( )	( )	Very good

9. **Did you have any ideas that are not from the table? Please list them if yes?**
Only for the teams with a supporting table.
Mark only one oval.

( ) Yes, what are the ideas_________________________
( ) No

Appendix H

Cognitive Social Structure Survey

Cognitive Social Structure BEAR

1. **Who would Emily go to for help during the development process?**
   Check all that apply.
   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

2. **Who would John go to for help during the development process?**
   Check all that apply.
   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

3. **Who would Taylor go to for help during the development process?**
   Check all that apply.
   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

4. **Who would Teresa go to for help during the development process?**
   Check all that apply.

   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

5. **Who would Sarah go to for help during the development process?**
   Check all that apply.

   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

6. **Who would Janelle go to for help during the development process?**
   Check all that apply.

   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

7. **Who would Emily share ideas with during the development process?**
   Check all that apply.

   - [ ] Emily
   - [ ] John
   - [ ] Taylor
   - [ ] Teresa
   - [ ] Sarah
   - [ ] Janelle

8. **Who would John share ideas with during the development process?**
Check all that apply.

- [ ] Emily
- [ ] John
- [ ] Taylor
- [ ] Teresa
- [ ] Sarah
- [ ] Janelle

9. **Who would Taylor share ideas with during the development process?**
Check all that apply.

- [ ] Emily
- [ ] John
- [ ] Taylor
- [ ] Teresa
- [ ] Sarah
- [ ] Janelle

10. **Who would Teresa share ideas with during the development process?**
Check all that apply.

- [ ] Emily
- [ ] John
- [ ] Taylor
- [ ] Teresa
- [ ] Sarah
- [ ] Janelle

11. **Who would Sarah share ideas with during the development process?**
Check all that apply.

- [ ] Emily
- [ ] John
- [ ] Taylor
- [ ] Teresa
- [ ] Sarah
- [ ] Janelle

12. **Who would Janelle share ideas with during the development process?**
Check all that apply.

- [ ] Emily
- [ ] John
- [ ] Taylor
- [ ] Teresa
- [ ] Sarah
- [ ] Janelle

13. **I am ________**
Check all that apply.

- [ ] Emily
- [ ] John
- [ ] Taylor
- [ ] Teresa
- [ ] Sarah
- [ ] Janelle

Appendix I

Dynamic Team Process Survey

App Development Process Survey

Your responses to this survey will be used is to understand the development process of the apps, the teamwork during the development process and to improve the course. Please respond to questions the best you can at the moment.

Your responses to this survey will not be inspected until after the course is over and grades have been assigned. Consequently, your responses will not affect your grade for the course.

We will ask you to take similar survives throughout the semester. Consequently, your MTU email is recorded so that we can track your responses across the different surveys.

Most of the questions are required accept a few at the end of the survey.

Your username (**wzhang5@mtu.edu**) will be recorded when you submit this form. Not **wzhang5**?
Sign out
* Required

1. **What are your goals for the app?** *

2. **Please rate your satisfaction of your app.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Not satisfied	◯	◯	◯	◯	◯	◯	◯	Very satisfied

3. **What are the important aspects of your app?** *
Please list all the aspects (at least 3) that you think are important, e.g. functionality, appearance.

4. **What aspects of your app's design are you currently satisfied with?** *
Please list all aspects (at least 3) of the app that you are satisfied with.

5. **What aspect of your app's design are you currently NOT satisfied with?** *
Please list all aspects of app that you are not satisfied with.

6. **What has influenced the development of your app?** *
Please list any influence (at least 3) of your app? e.g. teamwork, communication with scientist, design decision.

7. **Which of the above has most influenced the development of your app?** *
List only one of the above influences.

8. **Rate your enjoyment working with your team during your app development.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very much

9. **Rate your enjoyment working with your scientist during the your app development.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very much

10. **Were there any conflicts during the design or development of your app until now?** *
Mark only one oval.

⬭ Yes
⬭ No

11. **Rate the severity of the worst conflict, if there was one.**
Mark only one oval.

	1	2	3	4	5	6	7	
Very Bad	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Not Very Bad

12. **What do you think was the cause of the most severe conflict, if there was a conflict?**

13. **Rate the average CS team members' performance.**
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very good

14. **Rate the average HU team members' performance.**
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very good

15. **Rate the average Grad team members' performance.**
Mark only one oval.

	1	2	3	4	5	6	7	
Very bad	◯	◯	◯	◯	◯	◯	◯	Very good

☐ Send me a copy of my responses.