An experimental simulation of multi-site software development

N. Sadat Shami, Nathan Bos, Zach Wright, Susannah Hoch, Kam Yung Kuan,
Judy Olson and Gary Olson

Collaboratory for Research on Electronic Work (CREW)
Schools of Information and Business Administration, University of Michigan
701 Tappan Suite E2420
Ann Arbor, MI USA 48019
+1 734-647-7730
{ nsshami, serp, zwright, sushoch, kkuan, jsolson, gmo}@umich.edu

Abstract

Effective communication and coordination across multiple sites is extremely important for global software development. An experimental simulation that mimics an interdependent software division working across multiple locations was designed to study this phenomenon. Six experiments were run, each with participants divided into four or five sites depending on the availability of participants. We found that simulated workers at the same sites formed strong in-groups and were able to enlist help from their collocated colleagues at a much higher rate than from remote colleagues. These strong local in-groups inhibited cross-site collaboration. Remote workers had particular difficulty coordinating work with in-group members. However, certain isolated participants did perform well by being proactive in communication and reminding collaborators of reciprocal relationships. Future uses of this simulation will test different interventions such as instant messaging, calendaring, and traveling, which may help overcome the challenges of global software development and inform the design of future collaboration technologies.

1 Introduction

Software development is increasingly going global, with development occurring in multiple sites located in different geographic regions. Some of the reasons cited for this trend include improved information and communication technologies, increased globalization providing access to skilled labor at lower costs, the business advantage of capitalizing on the skills of workers knowledgeable about local market conditions and opportunities, and the need for flexibility to take advantage of mergers, acquisitions and alliances [23]. Companies have also begun experimenting with novel work arrangements such as 'Follow the sun' development to improve time to market. In such arrangements, software engineering teams make daily handoffs between remote sites across different time zones, allowing development to proceed nearly 24 hours a day. Furthermore, with eight out of ten large corporate IT departments projected to outsource offshore at least part of their technology services by 2004, the trend of global software development will continue [26].

Speed to market is a critical factor in succeeding in the highly competitive software industry [12]. In their studies of distributed software development teams, Herbsleb et al. found that cross-site communication and coordination issues cause a substantial loss of development speed [19, 20, 21]. Distributed items appeared to take about two and a half times longer to complete than similar items where all the work was collocated. Carmel showed that 58% of companies implementing 'Follow the sun' global software development methods saw the same or longer development times compared to traditional development methods [6]. Seaman & Basili in their study of participants in a software design and code review process at the IBM Toronto Software Lab found that interactions took longer and required more effort when participants were in physically distant locations compared to when they worked in close physical proximity to one another [39]. Prior empirical studies of large scale software development also illustrate the considerable amount of difficulty involved with communication and coordination [2, 11, 27, 41].

Difficulties arising out of working across distance also affect the inter-personal relationships among distributed workers. Software engineers involved in multi-site software development have to deal with the frustration of communicating with remote workers in different time zones, difficulties of language and culture, and intellectual property agreements that restrict communication. It may then come as no surprise that researchers have found that coworkers form less close relationships with distant colleagues, trust them less, and make uncharitable attributions about them [10, 19, 38].

Such real world evidence suggests that distance can be a formidable barrier to global software development. In the words of Olson & Olson, "The death of distance has been greatly exaggerated" [32]. However, as software development becomes more globally distributed, there are great rewards to be reaped for overcoming barriers to effective communication and coordination across multiple sites.

1.1 Prior research on managing communication and coordination across distance

Prior empirical studies of software development have found that software developers spend a considerable amount of time in communication and coordination activities [27, 37, 42]. These studies highlight the importance of communication and coordination in software development. Various approaches have been suggested to manage communication and coordination across distance. These include the introduction of communication technologies, explicit control mechanisms, and models for reducing the need for cross-site coordination.

With the aim of improving both formal and informal communication among distributed workers, communication technologies such as video [13, 15, 31], audio [25], chat [17], instant messaging [30] and text [7] have been introduced in the workplace. These technologies have had limited success and face challenges ranging from poor design to adoption. As a result, there is still much room for research on the design of such technologies.

Another approach to deal with the challenges of coordination across distance involves adopting practices such as the Capability Maturity Model [24, 36] and paying attention to software processes [4, 33]. These practices typically involve explicit and visible mechanisms for coordination such as planning, defining and following a process, carefully managing requirements and design specifications, measuring process characteristics, regular status meetings to track progress, implementing a work flow system and so on [24]. However, these plans and processes are only useful to the degree that one can anticipate various scenarios and contingencies. Factors that contribute to the highly variable nature of software development such as volatile requirements, unstable specifications, staff turnover etc. create limits on what can be reasonably planned and anticipated.

The final approach to managing distance has been to minimize cross site communication and coordination needs. Melvin Conway was one of the first people to recognize the relationship between product architecture and communication structure. He said that the structure of the product will mirror the structure of the organization that designed it – an idea that would later be known as Conway's Law [8]. David Parnas went on to clarify this relationship between product architecture and the people that design it. The notion of modularity, a cornerstone of software engineering, allows modules or 'work items' to be split in such a way that design decisions about each component can be made in isolation from decisions about other components [34]. Such division of labor reduces the requirement for coordination and communication across different parts of the organization involved in the design of different components. An implication of this would be to collocate the people involved with the development of the same component, since the tasks related with designing that component are tightly coupled and would require more communication. Tasks associated with different components are loosely coupled, and would require less communication. Grinter et al. studied six software engineering organizations where four models of minimizing cross-site communication and coordination were implemented. These models for dividing work among multiple sites were across functional areas of expertise, product structure, process steps and need for customization. The researchers found that regardless of what model was used, the complexities of communication and coordination of real world development projects led to problems such as conflicts between the central hub and remote sites, and with locating expertise [16].

Analyses of these approaches highlight the fact that technology is only a small part of enabling effective global teams. There is a need for tools and techniques that not only improve development processes, but also address organizational and social issues in global software development [28]. Consequently, a socio-technical frame of analysis may prove to be extremely useful in overcoming problems witnessed in empirical studies of global software development such as feelings of alienation among remote site workers, not knowing who to contact for help, lack of trust among workers at different sites etc.

1.2 The importance of cross-site social networks in global software development

When software developers working on the same project are collocated, there is ample opportunity for communication. A colleague can simply walk down the hallway to a coworker's cubicle, or engage in an informal conversation when they meet at the coffee machine. Being located in the same site creates such opportunities for collaboration and enlisting the help of colleagues. When development spans across multiple sites, such opportunities become more difficult because of the distance involved. There is a lack of shared context, a lack of cues regarding availability, and often a lack of trust for sharing information.

Due to the complex and volatile nature of software development, unexpected events often occur. When the resolution of these events requires coordination among multiple sites, this can be very disruptive. Developers report great difficulty in determining the appropriate person to contact at a remote site. Software developers often have no straight forward way of finding out who was responsible for a particular component on a remote site without resorting to workarounds [19]. In such situations, cross-site social networks become extremely important. Grinter et al. describe a situation where a problem with a large, complex component installed in a network in Mexico was solved by members of a UK site through calls to people they knew previously and maintained a relationship with [16]. As one of their respondents quotes 'things can be resolved over the phone if you have a good relationship with someone'. In the absence of such networks, the issue gets escalated to a manager. This could then potentially involve communicating with multiple people across multiple locations, and substantially impede resolution of the matter.

This value derived from a person's social networks is often referred to as 'social capital'. If we take two companies whose employees each have exactly the same profile of skills and knowledge, they would have identical intellectual capital. However, their performance might still differ radically from each other due to different levels of social capital. One company's employees are socially interconnected, and can use other employees' unique skills to their fullest extent, bringing the entire intellectual capital of the company to bear on the most important problems. Employees of the less-efficient company, by contrast, would confront every problem using their own skills, or perhaps those of workers who were institutionally or geographically close. The ability to use social capital to allocate intellectual resources efficiently has been referred to as the 'organizational advantage' [29]. This advantage is determined in part by the network of individual relationships that forms within a company, and also partly by the general climate of cooperativeness existing in the company.

In this paper we describe the use of an experimental simulation that mimics an interdependent software division working across multiple locations. We want to be able to study the effect of collocation, the presence of multiple 'sites' within a larger company, the integration of these disparate sites into larger group collaborations, and the influence of social networks in such collaboration.

2 Method

2.1 The Shape Factory Simulation

This paper reports data from use of the Shape Factory simulation environment. This experimental task was designed to study collaboration patterns among groups in many configurations across unequal media (face to face vs. computer mediated communication) and unequal geographic distribution (collocated vs. remote); much like when software development is spread across multiple sites.

The simulation is an online multi-player game. In order to allow participants to easily understand the simulation, we used the metaphor of a factory producing shapes – Shape Factory. In this simulation, each participant plays the role of a factory manufacturing colored shapes. For example, one participant might produce blue squares, and another green triangles. For each round, participants are given unique sets of 'orders' to fill. Orders are strings of shapes, e.g., 'purple circle-blue square-white diamond-red triangle'. Participants' effectiveness in the simulation is determined by their ability to get other participants to send scarce shapes in a timely manner for them to fill orders. Orders vary in length from 3-5 shapes, and their payoff increases with the square of their length, giving incentive to try to fill the longer, more difficult orders. Participants are distributed in different 'sites'. Some sites act as central hubs where several collocated participants play the simulation together in a room, while some sites only have single individuals playing the simulation in a single room, as shown in figure 1.

Participants engage in the simulation through laptop PC's with a wireless internet connection. Collocated participants are arranged around tables so that they can work on their laptops while conversing with each other. This initial arrangement of geographic distribution was chosen because it allowed us to look at the effects of collocation, the effects of multiple sites, and the effects of isolation within the same configuration.

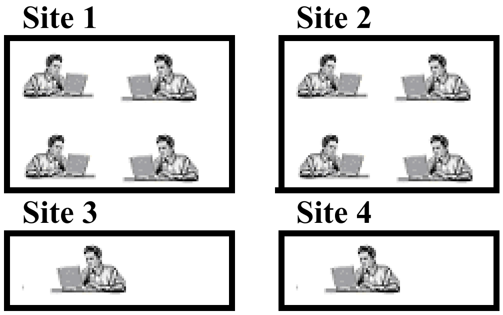

Figure 1: Configuration of collocated and isolated participants

Participants who are isolated can communicate with others only through the simulation's built-in messaging system, which is similar to email. Participants in Sites 1 and 2 also need to use the messaging system for cross-site conversations and for all official exchanges of shapes, but can talk around the table with other participants within their site. They can not visit other sites or verbally talk with participants from other sites.

This design of Shape Factory replicates some of the interdependencies and social network dynamics in distributed software development that we are interested in studying. We wanted a scenario where individuals are dependent to a certain degree on others for their success, similar to coordination situations in cross-site software development. In the simulation each participant has a need to identify potential collaborators and build and maintain relationships with them. They also have to operate with limited resources and time pressures. The simulation yields outcome measures of productivity that are meaningful at both the individual and at the group level.

Identify potential collaborators: In the simulation each participant has to decide who to collaborate with by making requests for shape(s) to fill their individual 'orders'. Participants are given more orders than they can fill each round, so they must choose which ones they want to assemble. Each order consists of shapes made by other participants in the simulation and is randomly generated from the set of active participant shapes. Participants choose which orders to go for based on their expectation that they can gain cooperation from those particular participants. Over time, participants may call on particular participants repeatedly, and ignore others. In a software development scenario, individual orders can be considered similar to a modification request (MR), where a request to enhance the functionality of the software or to fix a bug is made. Typically, the individual assigned as the MR owner needs to identify individuals with the necessary expertise and willingness to complete the task. These individuals may be located within the same site or across sites.

Need to build relationships: In order to fill orders and succeed in the simulation, participants must be able to obtain shapes from other participants. As a result, building and maintaining good relationships with other participants becomes very important. This is analogous to a MR owner being able to leverage his or her social network to enlist the help of other workers to complete the MR.

Limited resources: Participants' factories make only five new shapes per round, which is not enough to grant all incoming requests from other participants. So during each round, participants must allocate shapes to a limited set of collaborators, while also saving some for their own orders. This mimics the real-world condition that every worker has limited time, energy, and attention to give to others since they also have their own responsibilities to perform.

Limited time: Orders expire fairly quickly, and rounds of the simulation are kept short (15 minutes). A tax system also discourages participants from maintaining large inventories between rounds. Participants must thus choose collaborators based both on their expectation for cooperation, and promptness in replying. This mimics the real world in that the decision to collaborate may be influenced by speed and convenience of access.

Outcome measures: Individual participants' final scores are a measure of how well they were able to obtain shapes from other participants in a timely manner to fill orders. Group scores are also measures of whether a group was able to exchange its resources (shapes) efficiently and allocate shapes where they would do the most good. Since longer orders pay off higher proportional rates, groups and individuals that can coordinate delivery and production of longer chains score higher. Ineffective groups fill shorter chains, and may also lose many shapes to inventory taxes. So even though every same-size group is allocated the exact same number of shapes over the course of one 12-round simulation, groups vary widely on their scores. This is similar to real world organizations that may be identical in available talent but vary widely in effectiveness, due to differences in social capital.

2.2 Corollaries to 'real-world' collaboration

As in most simulations, Shape Factory is not intended to be a full simulation of multi-site software development, but rather is intended to reproduce some realistic dynamics of distributed work. The exchange of shapes between participants is intended to mimic the way that real-world collaborators exchange time, skilled work, and other favors. For example, programmers working for the same company would have different specialties, and part of their job would be to enlist help from relevant specialists. A highly efficient organization is one where employees can enlist and direct specialized skills of others without undue cost or delay. We have discussed previously how distributed software development can be inhibited when geographically separated workers fail to lend their specialized skills quickly. In Shape Factory, the scores of both individuals and the group are a reflection of their ability to enlist collaboration (through exchange of specialized shapes) when needed.

One criticism of the Shape Factory environment is that is it more competitive and less collaborative than most teamwork. This is probably true; in the tradition of experimental research, we have exaggerated the variables of interest in order to study them more effectively. But it is also true that even perfectly altruistic teams still face choices in how they allocate their time, attention, and favors. The interactions of the most well-meaning team may still be affected by the configuration of local and distant team members. And, in the workplace, performance reviews usually focus on individual versus collective performance.

As for the experimental conditions, we know that these also have some unrealistic aspects. Our isolated participants can communicate outside of their rooms only via text messaging. Real workers would at least have a phone available, and probably some ability to travel. For the purposes of this study it made sense to maximize the differences between groups. The point of the simulation is to study the effects of unequal media and unequal geographical distribution on social networks, which are very real challenges for distributed workers. Future experiments will add new variations, such as phone, instant messaging, calendaring, and travel, and measure the extent to which these variables ameliorate the effects of isolation.

We believe Shape Factory represents a paradigm shift toward more complex and large-scale experiments, which is needed to study phenomena that arise from real-world settings.

Field research is often seen as the remedy for the narrow scope of the laboratory study, but has its own limitations. Field work involves careful observation and analysis in situated settings, and thus avoids some dangers of oversimplification and decontextualization. However, fieldwork is incomplete because it is often too complex to be interpretable, and because it cannot be easily replicated. To take one example, Rocco, et. al. [38] found that in teams separated by distance, workers had lower levels of trust with remote colleagues compared to collocated colleagues. But this result is confounded, as most field studies are, by prior history between participants (the two divisions had joined the parent company at different times), extraneous variables (e.g. the two divisions were not exactly the same size), and particular circumstances (the two divisions had their own established cultures). Because of the complexity of the situation and the lack of replicability, it is very hard to know how widely to generalize the findings.

Software Engineering researchers such as David Parnas have stressed that neither field studies nor experiments alone can improve software development practices [35]. The research described in this paper is an attempt to create a middle layer research tool. Shape Factory is by necessity larger and less controlled than a typical experimental design. The effects we are looking for are emergent effects arising from many interacting participants, not direct effects of individual actions or intents. For example, we do not expect that any individual in the simulation will intend to ignore or marginalize participants located at other sites, but we expect that in a complex and fairly fast-paced simulation this effect will emerge as a result of many participants' actions. We also expect market forces to act in this larger group differently than they might in a smaller group experiment. These effects will emerge (if they do) because the simulation accommodates many individual choices and many different possible strategies. At the very least, the experimental paradigm described in this paper will be a useful addition to the set of methods researchers have at their disposal to study distributed collaboration.

2.3 Participants and procedures

We conducted six sessions of Shape Factory, involving 58 participants in all. Participants were recruited using a paid subject list, and were mostly undergraduate or graduate students. They were paid $35 each for their participation. After participants arrived at the lab, we asked them if they knew anyone else in their session. We excluded participants that did. Furthermore, since participants were identified using shape names rather than their real names, leveraging any prior acquaintance would be difficult. The sessions all followed roughly the same configuration of collocated and isolated participants. In each session, participants were divided into four-five sites depending on their availability. In two of the sites 3-4 participants worked collocated while the other two-three sites each had a single participant working isolated.

Participants were assigned to experimental location based on their order of arrival, with some correction in the last two experiments to ensure a gender-balanced group of isolated workers. Details of configurations are available in table 1.

	Site 1	**Site 2**	**Isolates**	**Total**
Session 1	3	3	2	8
Session 2	4	4	2	10
Session 3	4	4	3	11
Session 4	3	3	3	9
Session 5	4	4	2	10
Session 6	4	4	2	10

Table 1: Configurations of six Shape Factory sessions

The simulation was played in four-hour blocks, which allowed 12 rounds of 15 minutes each, plus 30 minutes for instructions, and two 15 minute breaks in the middle.

Participants were given simulation instructions and a self-administered quiz to test their understanding before the simulation began. They were told to pursue two goals: maximizing their own score, and maximizing the company score. The company included all participants in the simulation, and a running company total score was always visible on the interface next to their individual scores. Collocated participants were never told that maximizing in-site scores was a goal, although analysis will show that participants often behaved as if their in-site partners and them were a separate team.

3 Results

Statistical analysis of simulation results reveals two main effects: collocated participants did better than isolated participants, and collocated participants had an in-group bias toward trading with other participants in their own site. Additionally, message content reminding recipients of reciprocal relationships was found to be a statistically significant predictor of success.

3.1 Advantages of collocation

Participants who were collocated in a site with two or three other participants scored higher than isolated participants. In a regression analysis, being collocated predicted a significantly higher individual score in the simulation after the effect of individual sessions was controlled for using dummy variables (R2=.36, Beta=-.3, p<.01). This effect was also significant when we performed a t-test on z-scores not assuming equal variance.

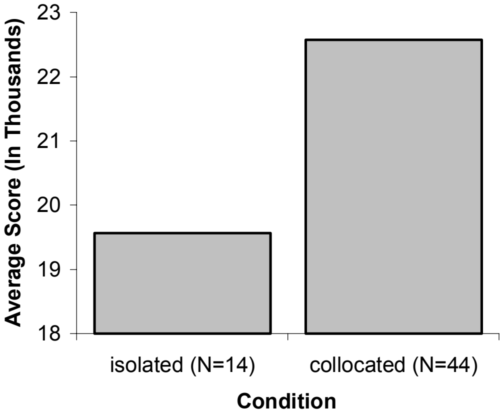

Figure 2: Mean individual score for isolated versus collocated individuals

Collocated participants could verbally request orders from each other, saving some time and effort, but the advantage of collocation went well beyond this. Although we did not record or systematically analyze verbal data from the collocated sites, simulation administrators did spend part of each simulation session in the back of one of the collocated sites listening to conversations. Participants were often observed reserving shapes for each other across multiple rounds, and arranging third-party trades with other sites (e.g. "blue circle says he'll send me one if he gets one from you"). They also exchanged information about other participants (e.g. "how responsive has green circle been to you?") and about simulation strategies. Collocated participants also reported that they chose which orders to try to fill based on which order had the most shapes from people within their site. These types of coordination efforts were very rarely done at a distance via the messaging system across sites, but were common among collocated sites.

3.2 In-group bias of collocated sites

Not surprisingly, analysis shows that the collocated sites on average traded shapes with each other more than with participants outside their site. Collocated participants sent each other participant in the site an average of 7.04 shapes over the course of the simulation, while sending only an average of 3.81 shapes to participants outside of their site, a significant difference (p<.001). They treated isolates and participants from the other collocated site about equally, averaging 3.76 and 3.81 shapes to each (no significant difference). Isolates showed no significant bias towards other isolates, sending them an average of 4.81 shapes, to 4.00 shapes to the average other participant.

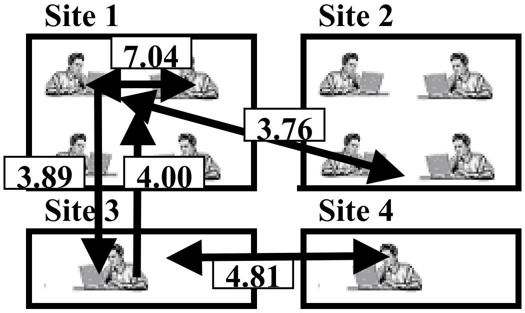

Figure 3. Average number of shapes exchanged between individual collocated and isolated participants, showing the large in-site bias between collocated participants.

While all participants received the same number of shapes over the course of a simulation session, not all delivered or received the same number, and isolates on average gave and received fewer.

When asked, most collocated participants reported that they chose which orders to fill based on which orders needed the most shapes from in-site participants. This indicates that the in-group bias resulted from choices made early on in each round – participants in collocated sites often did not even attempt to collaborate outside their site unless they had to.

The highly collaborative collocated sites, while providing a level of efficiency for site members, seem to have exacerbated the problem of remoteness for isolated participants. Analysis of the number of requests that were ignored by their recipients shows that collocated participants were significantly more likely to ignore incoming requests from outside their site, as compared with isolates. (R2=.173, Beta= -.371 p<.05).

Finally, collocated participants sometimes developed rivalries or animosities with other sites, irrespective of whether that site had collocated or isolated participants. In three different sessions we heard reports of makeshift 'embargos', where participants from one site would agree not to send shapes to one or more individual in another site, usually in retribution for a failure to help one of the group. These cross-site rivalries were not encouraged by the simulation instructions, which told participants to focus on the individual and whole-group goals, but emerged nonetheless.

3.3 Strategies of requesting from remote collaborators

We explored a number of possible explanations for why some participants did better than others in this simulation, looking in particular for strategies that would be helpful for working at a distance. We analyzed the record of shape requests through the messaging system. We excluded messages exchanged between collocated participants in this analysis for two reasons: first, because collocated participants did most of their negotiation verbally and so the text of their written messages was sparse and incomplete, and second because we were interested in isolating strategies useful for distant collaboration.

One aspect of request content was the strongest significant predictor of success in the simulation: references to past and future exchanges in the messages. These references included mentions of past cooperation ("I sent you a couple of shapes last round"), offers of immediate exchanges ("Let's trade shapes one for one") and promises of future cooperation ("I'll help you out when you need it"). For each participant we calculated a ratio of their use of these references to number of distant requests sent, and did an arcsine transformation to produce a normal distribution. These references significantly predicted participants' success in the simulation, as measured by their within-session z-score (R2=.23, Beta= 1.47, p<.01). These strategies were a stronger predictor of success than the sheer volume of requests sent, and when those two variables were entered into an equation together volume became non-significant, indicating that making exchange references was a more effective strategy predictor than sending larger numbers of requests.

We were unable to find statistical evidence for two other strategies that were suspected to be important: politeness and responsiveness. Several participants had told us that they thought 'politeness matters,' and felt that they had given preference to nicer requests, especially early in the simulation session. However, after coding messages for politeness, we were unable to find a relationship between this and success in the simulation. Responsiveness was also thought to be an important simulation strategy, especially for isolated participants. However, after analyzing both individuals' average delivery time and average response time, we were unable to find a relationship between this and success in the simulation.

4 Discussion

Participants of this simulation showed a strong tendency to preferentially collaborate with local rather than distant participants. These randomly-assigned participants with no prior history recreated the bias for local collaboration that has been shown in prior fieldwork studies, lending external validity to our study. Prior fieldwork has found a tendency for distributed teams to form subgroup identities based on location. Shared location provides a natural basis for the development of strong subgroup identities because such clusters of people typically enjoy more interaction and share more information with each other than they do with remote partners [10]. Armstrong and Cole studied software development teams and found that location was the basis for group identity among team members. They noted frequent instances where team members of the same site referred to each other as ‘us’ while referring to team members at other sites as ‘them’ [3]. Herbsleb and Grinter describe a competitive environment based on location that created subgroups in a software development team distributed between two sites in Germany and Britain. The fear that one site might be shut down and work consolidated in the other site led to weak information sharing across sites and strong in-groups within sites [19]. Herbsleb et al. also found that software developers feel they receive less help from their remote colleagues compared to their collocated colleagues [22]. Cramton studied students assigned to do a team project across national boundaries and found that team members were reluctant to interact with remote team members because of inconvenience factors such as the time and effort required [9]. This led them to increasingly interact with members that were only physically collocated with them. In our simulation, the in-group bias was demonstrated statistically by the fact that participants exchanged significantly more shapes with local participants than with distant participants. The advantages of coordinating with local individuals were quickly grasped by these participants. These advantages included more rapid coordination of exchanges and third-party brokering arrangements where a participant from a collocated site would get their collocated partners to send shapes to a participant in a remote site in exchange for the remote participant’s shape, essentially acting as the liaison between a collocated partner and the remote site. Such brokering arrangements are similar to the ‘contact person’ or ‘liaison’ between different remote sites mentioned in Herbsleb & Grinter’s study of distributed software development [19]. Being part of a collocated site was an advantage for individuals, and collocated participants achieved significantly higher scores than isolated participants. Collocation proved a powerful tool for improving productivity and encouraging rapid, flexible collaboration, as shown in prior studies [38, 40, 42]. The ease of coordination and richness of information exchanged demonstrates the power of collocated work.

This simulation also starts to illustrate some of the disadvantages that cohesive collocated groups may present for the larger organization. Collocated participants were significantly more likely to ignore incoming requests from outsiders than were a control group of isolated participants. We also observed collocated participants exchanging third-party information that sometimes took the tone of gossip, and enacting group embargoes against specific other participants in different sites. Our findings are similar to findings of Herbsleb & Mockus that cross-site relationships have less of a team orientation and are less mutually beneficial than same site relationships [20].

These observations illustrate the challenges of integrating cohesive, collocated sites into the larger organization. Distributed organizations must learn to use these highly efficient collocated sites without creating cliques or inter-division rivalries that may be a drag on the overall effectiveness of the organization. Cohesive sites need to develop means of working cooperatively with other sites and including and integrating isolated workers.

In our simulation, isolated participants had significantly lower overall scores than their collocated counterparts. Arranging trades was more difficult for them because they could use only the messaging system to communicate and build trade relationships. They also suffered exclusion from the in-group trade cliques that formed. As the collocated sites engaged in heavy in-group trading and deal-making, the isolates were cut out of the loop, and were less effective as a result.

On a more positive note, however, there were a few isolated participants that were able to do well in the simulation despite their disadvantage. This included two participants who took second place in their respective sessions, and one who outperformed all others in her session by a large margin (4,300 points, or .82 standard deviations). Based on these results, it may be fair to conclude that it is possible to operate effectively as an isolate, albeit more difficult.

We examined some of the strategies that seemed to make isolated participants more effective. As noted before, explicitly referring to the past, present, and future exchange was positively correlated with success. This was true for all distant communications between all participants, and it may have been especially important for isolated participants for which all communication was distant.

We were unable to demonstrate statistically the effect of two other effects that isolated participants reported as being important: politeness and responsiveness. But we know that the isolated participants that were considered dependable tended to respond quickly to requests, and suspect that responsiveness does have an important role in effectiveness. High-scoring isolates were also very proactive in their communications, sending lots of requests, usually very early in a round.

There already exists in the trade literature recommendations for how to integrate distributed teams [14] and include remote employees [18]. Some common advice includes the use of availability and awareness tools, using richer media, encouraging travel, and using semi-permanent liaisons between divisions. These ideas exist mostly as reasonable-sounding recommendations unsupported by empirical data. The effectiveness of these types of interventions has typically been difficult to measure, and it has been almost impossible to compare proposed interventions to each other, because of the difficulties inherent in field work described earlier. Now that we have successfully replicated some of these phenomena in a lab setting, we hope that Shape Factory and similar simulations will provide a testing ground for many and various proposed interventions into distributed work. Testing interventions and observing how people react to them will inform the design of future collaboration technologies in support of global software development. It is in this use of the simulation that we anticipate the greatest research rewards, and the study described in this paper is an important first step towards that.

5 Conclusion and future directions

In this paper we highlighted the importance of cross-site social networks in global software development teams to leverage the benefits of social capital. Through the use of a novel experimental paradigm, we found that participants with no prior history formed strong in-group biases towards participants that were collocated with them. These strong in-groups inhibited cross-site collaboration leading to inefficient allocation of resources (shapes) and lower company scores. Findings from our study are similar to prior field studies of real world global software development teams.

Findings from this first run of Shape Factory have motivated three planned follow-up studies which will clarify some points related to distributed sites. The first line of study will focus on manipulating the configuration of sites by having single individuals in the majority of sites, to see whether isolates perform better when there are fewer cohesive collocated sites and being isolated is the norm. The second line of research will focus on giving remote participants more communication choices – phone, instant messaging, calendaring, email broadcasting, etc. to see whether they can build relationships with distant colleagues more easily with more tools. The third line of research will give remote participants some training in effective strategies, and see whether the barrier of distance can be overcome with proactive communication and responsiveness.

We are optimistic that looking at issues surrounding global software development from a socio-technical lens will help organizations struggling with the challenges of communication and coordination created by distance. Indeed solutions that allow us to build relationships with distant colleagues as easily as we build relationships with our local colleagues will go a long way in alleviating the challenges of distance.

Acknowledgements

In the course of this research we received invaluable advice and feedback from Beth Veinott, Dan Horn, John Miller and Wei Huang. We are also grateful to the staff at the School of Information for aiding us in the considerable logistical challenges involved in this research. This research was supported by NSF grant IIS 9977923 to the Olsons.

About the Authors

N. Sadat Shami will start his PhD in Information Science at Cornell University starting Fall 2004. He has a Master's degree in Information from the University of Michigan and is a Research Associate at the Collaboratory for Research on Electronic Work.

Nathan Bos is Assistant Research Professor at the School of Information and a Research Scientist at the Collaboratory for Research on Electronic Work at the University of Michigan.

Zach Wright holds a Master's degree in Human Computer Interaction from the School of Information at the University of Michigan. He currently works for the Visible Human Project at the university.

Susannah Hoch holds a Master's degree in Human Computer Interaction from the School of Information at the University of Michigan. She currently works as a Human Systems Engineer at Aptima Inc.

Kam Yung Kuan is a PhD student at the University of Michigan Business School.

Judy Olson is the Richard W. Pew Collegiate Professor of Human Computer Interaction at the School of Information at the University of Michigan. She is also a Professor at the Business School and Department of Psychology at the university.

Gary Olson is the Paul M. Fitts Collegiate Professor of Human Computer Interaction at the School of Information at the University of Michigan. He is also the Associate Dean for Research at the School of Information.

References

[1] M.J. Abel, "Experiences in an Exploratory Distributed Organization," *Intellectual Teamwork: Social Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, eds., pp. 489-510, 1990.

[2] T.J. Allen, *Managing the Flow of Technology.* MIT Press, 1977.

[3] D. Armstrong, and P. Cole, "Managing distances and differences in geographically distributed work groups", in *Distributed work: New ways of working across distance using technology* (P. Hinds & S. Kiesler, Eds.), MIT Press, 2002.

[4] B. Boehm, "Anchoring the Software process," *IEEE Software*, vol. 13, no. 4, pp. 73, 1996.

[5] F.P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering.* Addison Wesley, 1995.

[6] E. Carmel, *Global Software Teams*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[7] E.F. Churchill and S. Bly, "It's all in the words: Supporting work activities with lightweight tools," *Proc. Int'l ACM SIGROUP Conf. Supporting Group Work*, pp. 40-49, 1999.

[8] M.E. Conway, "How Do Committees Invent?" *Datamation*, vol. 14, no. 4, pp. 28-31, 1968.

[9] C. Cramton, "The mutual knowledge problem and its consequences for dispersed collaboration," *Organization Science, 12,* 346 – 371, 2001.

[10] C. Cramton, "Attribution in distributed work groups," in *Distributed work: New ways of working across distance using technology* (P. Hinds & S. Kiesler, Eds.), MIT Press, 2002.

[11] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM,* vol. 31, iss. 11, pp. 1268-1287, November 1988.

[12] M. A. Cusumano, and D. B. Yoffie, *Competing on Internet Time,* New York, NY: The Free Press, 1998.

[13] P. Dourish and S. Bly, "Portholes: Supporting Awareness in a Distributed Work Group," *Proc. ACM Conf. Human Factors in Computing Systems*, pp. 541-547, 1992.

[14] M. Dziak, and G. Gordon, *Telecommuting Success: A Practical Guide for Staying in the Loop while Working Away from the Office.* JIST Publishing, 2001.

[15] R.S. Fish, R.E. Kraut, and R.W. Root, "Evaluating Video as a Technology for Informal Communication," *Proc. ACM Conf. Human Factors in Computing Systems*, pp. 37-48, 1992.

[16] R.E. Grinter, J.D. Herbsleb, and D.E. Perry, "The Geography of Coordination: Dealing with Distance in R & D Work," *Proc. Int'l ACM SIGROUP Conf. Supporting Group Work*, pp. 306-315, 1999.

[17] M. Handel, and J.D. Herbsleb, "What is chat doing in the workplace?," *In Proceedings of CSCW* , pp. 1-10, 2002.

[18] M. Haywood, *Managing Virtual Teams: Practical Techniques for High-Technology Project Managers,* Artech House, 1998.

[19] J.D. Herbsleb and R.E. Grinter, "Splitting the Organization and Integrating the Code: Conway's Law Revisited," *Proc. Int'l Conf. Software Eng.*, pp. 85-95, 1999.

[20] J.D. Herbsleb, and A. Mockus, "An Empirical Study of Speed and Communication in Globally Distributed Software Development," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, June 2003.

[21] J.D. Herbsleb, A. Mockus, T.A. Finholt, and R.E. Grinter, “An Empirical Study of Global Software Development: Distance and Speed,” *Proc. Int’l Conf. Software Eng.*, pp. 81-90, 2001.

[22] J.D. Herbsleb, A. Mockus, T.A. Finholt, and R.E. Grinter, “Distance, dependencies, and delays in a global collaboration," *Proceedings of Computer Supported Cooperative Work.* New York: ACM, 2000.

[23] J.D. Herbsleb and D. Moitra, “Global Software Development,” *IEEE Software*, vol. 18, no. 2, Mar./Apr. 2001.

[24] J.D. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes and M. Paulk, “Software Quality and the Capability Maturity Model,” *Communications of the ACM*, Vol. 40, No. 6, pp. 30-40, 1997.

[25] D. Hindus, M.S. Ackerman, S. Mainwaring, and B. Starr, “Thunderwire: A Field Study of an Audio-Only Media Space,” *Proc. Computer Supported Cooperative Work Conf.*, pp. 238-247, 1996.

[26] J. King, "IT's Global Itinerary: Offshore Outsourcing is Inevitable," *Computerworld*, vol. 37, iss. 37, pg. 26, Sept. 15, 2003.

[27] R.E. Kraut, and L. Streeter, “Coordination in Software Development,” *Communications of the ACM,* vol. 38 iss. 3, pp. 69-81, March 1995.

[28] F. Lanubile, D. Damian, and H.L. Oppenheimer, "Global Software Development: Technical, Organizational, and Social Challenges," *ACM SIGSOFT Software Engineering Notes,* vol. 28, iss. 6, November 2003.

[29] J. Nahapiet, and S. Ghosal, "Social capital, intellectual capital, and the organizational advantage," *Academy of Management Review*, April 1998.

[30] B.A. Nardi, S. Whittaker, and E. Bradner, “Interaction and Outeraction: Instant Messaging in Action,” *Proc. ACM Conf. CSCW,* pp. 79-88, 2000.

[31] A. Obata and K. Sasaki, “OfficeWalker: A Virtual Visiting System Based on Proxemics,” *Proc. ACM Conf. CSCW*, pp. 1-10, 1998.

[32] G.M. Olson, and J.S. Olson, "Distance Matters," *Human Computer Interaction*, 15, 139-179, 2001.

[33] L. Osterweil, “Software Processes are Processes Too,” *Proc. 9th Conference on Software Engineering,* 1986.

[34] D.L. Parnas, “On the Criteria to Be Used in Decomposing Systems into Modules,” *Comm. of the ACM,* vol. 15, no. 12, 1972.

[35] D.L. Parnas, “The limits of empirical studies of software engineering,” *Proceedings of the 2003 International Symposium on Empirical Software Engineering,* IEEE Computer Society 2003.

[36] M. Paulk, B. Curtis, M. Chrissis and C. Weber, *Capability Maturity Model for Software (Version I. I)*, Technical Report, CMU/SEI-93-TR-024, Pittsburgh, Software Engineering Institute, Carnegie Mellon University, February, 1993.

[37] D.E. Perry, N.A. Staudenmayer, and L.G. Votta, “People, Organizations, and Process Improvement,” *IEEE Software*, vol. 11, no. 4, pp. 36-45, 1994.

[38] E. Rocco, T.A. Finholt, E.C. Hofer, and J.D. Herbsleb, "Out of sight, short of trust", *Presentation at the Founding Conference of the European Academy of Management.* Barcelona, Spain, April 2001.

[39] C. B. Seaman and V. R. Basili, “Communication and organization in software development: An empirical study,” *IBM Systems Journal*, vol 36, number 4, 1997.

[40] S.D. Teasley, L. Covi, M.S. Krishnan, and J.S. Olson, "How does radical collocation help a team succeed?," *Proceedings of Computer-supported cooperative work.* New York: ACM, 2000.

[41] D.B. Walz, J.J. Elam, and B. Curtis, "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration," *Communications of the ACM,* vol. 36, iss. 10, pp. 62-77, October 1993.

[42] J. Wu, T.C.N. Graham, P.W. Smith, “A Study of Collaboration in Software Design,” *Proceedings of the 2003 International Symposium on Empirical Software Engineering,* IEEE Computer Society 2003.