Understanding the Collaboration Needs of a Developing Distributed Team

Patrick Wagstrom, Laura Dabbish, Anita Sarma, and James Herbsleb
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, 15213
{pwagstro, dabbish, antz, jdh}@cs.cmu.edu

ABSTRACT
The initial period of team development and formation can be a tumultuous period when communication patterns and work practices are established that will remain with the team for a long period. Therefore, it is particularly important that teams establish practices and adopt tools that will facilitate the long term success of the team. In this paper we examine a developing distributed team in process of adopting tools to support distributed work. We identify initial tool selection decisions made and elicit the requirements for future tool use from the team. We then relate these requirements to existing tools and research and provide recommendations for fertile research areas in the field of CSCW.

Author Keywords
Distributed teams, CSCW tools, collaboration, coordination

INTRODUCTION
Teams, even when fully co-located, rarely begin life as fully functional entities. An evolutionary process takes place that guides a team from a collection of individuals with varying goals and roles into a cohesive focused entity, such as the "Forming - Storming - Norming - Performing" progression [15]. Modern distributed teams often face many of the same challenges as co-located teams, but often must leap additional hurdles challenges due to time lag between members, missing communication context, and tool standardization and distance between team members [12, 13].

A large amount of previous research in CSCW related fields has addressed ways that technical solutions can help distributed teams and also present additional challenges. Amongst these solutions are text chat [7, 10], real time video conferencing [9, 11], knowledge management [1, 6, 14], and shared calendaring [16]. However, even with these technologies distributed teams projects still face an uphill battle to overcome the difficulties of distributed work [2,15].

Design of these tools has typically proceeded with an eye toward features for distributed teams that have already adopted a set of norms and have established communication patterns. Adoption and full utilization of these tools never occurs overnight, rather as the team evolves and matures, the practices around the tools evolve allow the team to become more effective at overcoming distance. This transition and evolution period can take many months or even years, during which time the team is more likely to rely on connections with co-located individuals than they are at later stages of team development [3], leading to the potential for underutilization of the tools. In addition to relying on local connections for coordination issues, early stages of team development frequently sees many opportunities for tool exploration and experimental tool deployments. When combined with the reliance on local links, this results in pockets of tool usage within a team where different sub-groups utilize different work practices and possibly incompatible tools. Like defects in software engineering, normalizing and resolving tool incompatibilities early in the team formation process may incur little cost while waiting for a later point in time, after work practices are established and information is stored in many incompatible tools, combined with the distributed nature of the team will result in a much greater cost [8].

Another issue that teams face while standardizing on new tools is how to introduce new team members to the project. Once a team has formally adopted tools it is clear where to point new team members for information. For example, on a team with only a single wiki or a single location for all mailing lists, new team members need only visit a single location to find the necessary information. In contrast, a developing team with disparate resources may frequently have information silos that are not accessible to new team members. This results in missed communication and slows the rate of information diffusion within the team [1, 17].

The issues raise the question of how we can ensure CSCW technology is effectively deployed to developing teams in a manner that allows the teams to get maximum usefulness from the tools, address the needs of the team now and into the future, and are feasible solutions for a young team. To understand these issues, we analyze a large engineering project that is still growing, establishing norms, and seeking tools to use for team collaboration. We base our research on four months of intensive observations of team meetings, communications, presentations, and interviews with team members. We begin with a description of the team, its members, and current state of team formation. We then provide an overview of the tools used by the team prior to our engagement with the team. Next we describe the communication and coordination needs of the team as identified by the team and through our observations of team members. Finally, we describe a set of proposed solutions for this developing team based the availability and practicality of implemented CSCW technology for distributed teams in a real world environment.

TEAM BACKGROUND

The subject of our research is a team competing in the Google Lunar X-Prize (http://www.googlelunarxprize.org/) – a competitive challenge funded by Google that awards a $30 million prize to the first privately funded team to land a robotic rover on the moon that can travel 500 meters and transmit high definition video back to Earth. This challenge has attracted a number of competitors from around the world ranging from hobbyists to universities to small commercial startups. Our focus is on a single team, Astrobotic, which is particularly well suited to the challenge and is considered one of the front runners to win the prize. Functionally, the team has broken the project into several core areas: funding and business, lunar rover, lunar lander, and mission operations. The team is composed of four major players, each with unique skills in one of these areas that makes Astrobotic ideally suited for victory in this challenge.

At the core of the team is Carnegie Mellon University in Pittsburgh who are coordinating all the teams, performing engineering for the lunar rover, and doing much of the integration work between the different components. This team also acts as an effective back stop when other members of the project face external pressures that limit their roles. There are approximately 20 graduate student and researchers working full time within this group – many of whom have had previous success through the DARPA Urban Challenge which saw their autonomous vehicle successfully navigate an urban environment to win a $2 million prize from DARPA. Most of the team members are located in Pittsburgh, however members frequently travel and a few of the members live primarily in other states, forcing them to collaborate from afar. The team lead, Red Whittaker, is one of the foremost roboticists in the world, having previously worked to send robots to adverse environments such as the core of nuclear reactors and the remote deserts of Antarctica. In addition to the full time members, the university also runs a class each semester which cycles students in and out of the project on a temporary basis and is in a constant recruitment mode looking for skilled individuals to fill team needs. In this way, the membership of the team at Carnegie Mellon is constantly in flux with the 20 full time members forming a solid core.

Working closely with the team at Carnegie Mellon is Astrobotic Technologies – a startup firm that seeks to raise funds for the engineering, design, and launch of the vehicle and explore the possibilities for business ventures based on robotic missions to the moon. This team was originally distributed with most members working from the west coast of the United States and periodically traveling to Pittsburgh for face to face meetings. More recently this team has relocated full time to Pittsburgh in an effort to consolidate the major components of the team in a single city. The team behind Astrobotic Technologies is relatively small, with fewer than 10 primary individuals working on the business plan and marketing of the venture. In contrast to the team at Carnegie Mellon, as a business venture the membership of Astrobotic Technologies is quite stable.

While Carnegie Mellon has extensive experience with the creation of robots that function in adverse environments, there is little experience landing vehicles on extra-planetary surfaces within the team. To fill in this need, the team added members from the Lunar and Planetary Lab (LPL) at the University of Arizona and Raytheon, both located in Tucson. This portion of the team has extensive successful experience with landing and operating vehicles millions of miles away. Their success has been publicly demonstrated both through the wildly successful Cassini/Huygens probes that penetrated the atmosphere of Saturn's moon Titan and the current Mars Polar Lander, which discovered water based ice near the surface of the Martian North Pole. The team at Raytheon typically numbers around five experienced engineers, while the team at the LPL ranges between five and ten. Like Carnegie Mellon, the University of Arizona also runs a semester long class for students interested in participating in the design of the lunar lander that brings in about 20 students per semester.

Finally, as this is a new and novel engineering project – the likes of which only the United States government has previously performed, the team relies extensively on a network of external collaborators and consultants to address particular technical issues that arise in the course of design and testing. These external consultants are typically only involved for a brief period, such as a few short teleconference calls particular to their area of expertise, but it is believed that in the future these consultants will become more valuable to the team and their use will increase.

The team utilizes a variety of meetings as their primary methods of collaboration. The team at Carnegie Mellon holds weekly "All Hands" meetings that are organized by functional area of the rover design. Frequently these meetings have members who are calling into teleconference from remote locations. Often employees and management of Astrobotic Technologies participate on the phone or in person and occasionally individuals call in from the teams at LPL and Raytheon, however this is unusual given the meeting's normal scheduling at 9am Pittsburgh time (6am Tucson time). These meetings typically have only a loose agenda and consist primarily of engineering updates, although they frequently segue into business development and product management issues. A typical "All Hands" meeting will have between 15 and 25 people present and last between 75 and 120 minutes.

In addition, on a weekly or bi-weekly basis there are management conference calls between the principles team members at Carnegie Mellon, Astrobotic, LPL, and Raytheon. These meetings are usually jointly managed by an individual at Carnegie Mellon and another individual at Raytheon. The major focus of these meetings is on cross site issues, project vision, and project funding issues. These 30 to 60 minute calls attract between five and ten participants, although the Carnegie Mellon team takes the calls in a project work room which allows project engineers to spontaneously contribute when needed.

At various stages in the design and implementation of the project, the team holds "Design Reviews" – 2-5 hour long meetings that allow each component of the project to present a complete status update of their component. These meetings are typically used as an opportunity to collaborate closely with one another and obtain feedback on the progress of the design and integration for the rover and lander. Depending on the stage of the review, these reviews may take place all at a single site, or may have engineers and project managers calling in from other sites. End of the semester project presentations from the classes at Carnegie Mellon and the University of Arizona also provide an opportunity for feedback on the current design and implementation of the project and often feature individuals from multiple sites participating.

Beyond these regularly scheduled meetings, the team also utilizes many meetings that either are one time events, or occur only for a short period. For example, there was short period where there was a weekly composites teleconference between Carnegie Mellon and LPL. Individuals and teams also frequently schedule meetings with much shorter notice to address particular needs that have arisen.

For our research, we observed and video taped all regularly scheduled meetings and design reviews for a period of four months. Opportunistically we were able to observe and video tape other meetings. The result was between two and four observations a week taking between two and eight hours a week of observation time. Video recordings were synchronized with the notes of the observers and a selection of the meetings have been fully transcoded. We augmented these observations through a series of 28 interviews with project members. These interviews were focused on gaining a technical understanding of the project and understanding the complete work and collaboration processes of the team.

Our participation in the project began as observers, but has moved into that of true ethnographers and participants. When members of the team found out the backgrounds of our group of observers as software developers and electrical engineers we began to see questions about project engineering directed to us. This level of intimate involvement allows us to better understand the true requirements for communication and collaboration within the project. In addition, as the project has grown and matured, the coordination needs have become more complex and we have periodically met with team members and project managers to help solve these issues and implement solutions.

INITIAL TOOLS IN USE

Prior to our observations, the team had already adopted a variety of different tools for coordination and collaboration, and these tools varied across sites. The team at Carnegie Mellon had previously utilized BaseCamp, a subscription service from 37 Signals, for engineering projects and it came as no surprise they chose to once again use the service. The service is designed for group collaboration and functions as a slightly enhanced message board. Individuals can post messages and files to a general message board where all team members can read and join in the discussion. They also have the option of specifying a set of recipients when the message is created, in which case those individuals will also receive a copy of the message in their email. The tool allows individuals to be clustered into mutually exclusive groups; in this case individuals were placed in groups according to their formal affiliation - Carnegie Mellon, Raytheon, University of Arizona, or Astrobotic Technologies. In addition, there were sub areas created for the classes taught at Carnegie Mellon and the University of Arizona, which allowed the students to collaborate. Although this tool is primarily used by Carnegie Mellon, individuals from LPL, Raytheon, and Astrobotic use it heavily.

In addition to BaseCamp, the team at Carnegie Mellon utilizes a number of mailing lists – hosted both at the university and on Astrobotic servers. These mailing lists see only occasional usage and many of the lists have never been used. Different sub-teams used the system in different ways – for example the team developing the software used mailing lists extensively for status updates and automated testing messages. In contrast, the team working on the body of the lunar rover never used the mailing lists, despite there being a number of different lists to address needs. There were also several different general purpose mailing lists which were sometimes used, but it was clear that not all team members received mail from these lists. It was unclear who had the responsibility of updating the membership of the general mailing lists which frequently meant that newer members did not receive emails sent to the list. Combined with the fact that some lists were only periodically used, it could be several weeks before an individual discovered they were missing meetings because there were left off a critical mailing list.

Individuals within the team had also taken the initiative to set up collaboration tools they believed to be useful to the team. One individual set up a MediaWiki installation to document as much of the project as possible. The software team had an installation of Bugzilla to track software defects and one of the project managers utilized a commercial tool, FRACAS, to manage non-software related defects. In all cases these tools were never utilized beyond the individuals or subgroups that initially set them up despite the fact they may contain large amounts of useful information, such as the project Wiki. In addition, none of these tools integrated with any of the other tools already in use, requiring each user to obtain a new login and password combination from the team member responsible for the tool.

The team at Astrobotic Technologies previously had expertise with Microsoft tools and has deployed a Microsoft SharePoint and Microsoft Exchange solution. This tool is used primarily to coordinate the business aspects of the Astrobotic commercial venture and is not used or accessible to individuals outside of Astrobotic Technologies. From our observations this restriction was never a hindrance to the project engineers at other sites.

The team at LPL and Raytheon utilizes yet another solution for content management, an installation of Knowledge Tree, a software package that is very similar to BaseCamp, with the exception that it is Open Source and an installation requires no recurring fees. All individuals are given accounts on the Knowledge Tree installation, however in practice it is only lightly used even within the teams in Tucson. Most email messages from LPL and Raytheon went through the BaseCamp installation provided by Carnegie Mellon.

Upon the start of our observations we were given access to all systems with the exception of the SharePoint and Exchange Servers hosted by Astrobotic Technologies. This allowed us to first hand evaluate the number of different communication systems that the team used. All together we observed 14 different communication systems in use by the team, many of which duplicated functionality but were run by different sub-teams.

As we began to catalog all of the communication tools in use by the team it became apparent that the current ad-hoc infrastructure adopted by the team would not support the team's long term success. Many team members, even those in management, were not fully aware of all of the communication tools in use by the team and were concerned that critical pieces of information may never reach their appropriate destinations or be properly archived. Furthermore, with no central individual responsible for team tool selection, when an individual would deploy a tool they believed necessary for project success it was very difficult to get other team members to use the tool. At this point, we began to work with the team to identify the actual needs of the growing team, and analyze how current generation tools address or fail to address the needs of such a dynamic effort.

COMMUNICATION NEEDS

We began our analysis of team needs by first performing a survey of the existing tools and identifying the communication needs that arose in our observation of team meetings and interactions. We then shared this list with several project managers who helped augment the list, and finally had a meeting with several of the team members who were already maintaining collaboration systems for the team where we formalized the needs more and identified possible solutions. Primarily the needs for the team were, in approximate order of importance, seen as facilitating shared meetings, accessible mailing lists, robust document repositories, support for presenters and participants during meetings, complete integration across tools, ease of use and installation, and cross platform functionality.

Shared Meeting Times and Calendaring

There are several different challenges that face the team when scheduling meetings. They face some of the usual challenges of distributed teams encompassing multiple organizations. The standard time difference between the two major sites is three hours, resulting in numerous meetings that occur over lunch time or outside of normal 9-5 business hours. This is a constant difference that most of the team members have learned to deal with. A much more difficult problem is that of differing institutional schedules. Being located at two universities results in many members being temporarily pulled away for other commitments, such as classes at the start of a semester or university holidays.

The team has several standing meetings that occur on a regular basis, including the weekly "all hands" meeting that features every member at the primary site, and a bi-weekly management conference call featuring various project leads at the primary and remote sites. In addition to these meeting times, individuals or project leads set up periodic meetings to cover a wide variety of issues from technical integration to testing results. In most cases the scheduled meetings are held in the same location from one meeting to another and there is one individual who is in charge of meeting logistics - such as setting up the conference call if needing, collecting and assembling all slides, and managing the meeting. The meetings that occur reliably and are only at a single site rarely have any problems with scheduling. However, the management conference calls often pose issues as it can be unclear if the call is happening each week. Typically if the call is not going to happen one of the leads sends out an email via the BaseCamp system, but often individuals fail to receive the update in time to modify their schedule. The problem becomes worse for irregularly scheduled meetings. Often these meetings are cancelled or moved with only an email being sent to the participants and nothing posted to BaseCamp, leaving some peripheral participants confused about the meeting status.

To alleviate these problems, the team seeks a tool that allows all participants to view and edit a shared calendar. A key feature of the calendar is that it must allow email notification of meeting change status, allowing team members with PDA cell phones to receive notification of meeting changes away from their computer. Another desired feature is to be able to invite individuals to a meeting and have it automatically appear on their calendar. The team had a lightly used Google calendar which was able to meet some of these needs, but it was not able to integrate and easily provide free/busy information for all team members. Furthermore, team members had difficulty with some of the email notification and invitation features of the tool. Other commercial groupware systems such as Microsoft Exchange and Lotus Notes provide these features as standard within the tool; however the use of these tools would be too radical of a change to the workflow for many team members.

Mailing Lists

Similar to most teams, both co-located and distributed, the team relies heavily on email for communication and coordination. However, the way that email is used is not consistent between team members at a single site, much less across site. Largely this has to deal with the evolution of tools within the team and experiences of individual team members with mailing lists. Members of the team responsible for developing software for the rover had extensive experience using email for both communication and automated status reports - such as messages when a build of the software was completed. These members have a traditional view of what a mailing list is, and it corresponds very closely to how most mailing list software functions; a message to a mailing list goes to a set group of people and is archived for later reference through a web interface. These members also typically preferred to read their email through a standard email client where they could set up automated filters to help sort and categorize incoming mail. One concern raised by the administrator of the project mailing lists, however, was the immense administrative effort required to keep all mailing lists up to date. While the mailing list infrastructure provided an interface for members to add themselves to lists, many members would simply ask the administrator to add them rather than learning how to utilize the web interface for list subscription and management.

Other team members had very different views of how mailing lists should function and believed that the current web forum type setup provided by BaseCamp was sufficient for the team. They highlighted the fact that it was easy to create an ad-hoc mailing list for individuals that might be interested in a particular message such that only that subset of individuals would receive the message as an email and the rest of the team could easily see it through the standard web interface. Individuals receiving the email could reply to the message and it would be sent to all other members of the ad-hoc list. Members emphasized that this functionality was needed when cross-cutting engineering concerns arose or there was a need to collaborate across functional areas; something that more traditional mailing list software could not do. However, when pressed they conceded that the creation of ad-hoc lists frequently omitted individuals who may find the information useful.

In reality, the desires of both groups were similar; however they examined the situation from very different perspectives. The issue is that in a developing team it must be easy to create ad-hoc groups in addition to static groups. Individuals must be able to be in multiple groups at the same time - something not easily possible in BaseCamp, and the membership of these groups should be made overtly public - something that is typically not available in most mailing list software. In addition, the messages sent to these groups should be made public and archived in a way that is searchable for future reference.

Shared Document Repositories

Most teams begin their collaboration by passing most files back and forth through email. For small teams and short term projects with relatively few documents shared between individuals, this works well -- particularly in the co-located case where face-to-face contact reinforces relationships and helps to ensure that individuals are not inadvertently forgotten when a new version of a document is created. However, as a team grows sharing documents via such a medium comes under attack from a variety of angles: passing a single file back and forth many times leads to confusion about the newest version of the file, archiving of files is nearly impossible as individuals are forced to search through their mailboxes for files, large files run into issues with maximum email attachment size, and changing roles in the team may leave others without access to critical documents. Furthermore, due to limitations inherent in the HTTP protocol used for most web based systems, uploading of large files necessary for many engineering projects, which may be hundreds of megabytes in size, is not practical.

As work progressed within the team, they had already begun to outgrow many of the document repository solutions they were using. The team invested significant effort in looking at alternative web based systems, such as what is provided by Google Apps for Domains, before ruling it out because of small (10MB) file size limits. Furthermore, most of these tools created disconnects in workflow that made it difficult to ensure that files in the repository were the most up to date. For example, if an engineer would modify a large CAD diagram of a new part and forget to synchronize the file to a repository, the next day when another engineer seeks to route electrical cables through the part they would be working off an obsolete version of the document without being aware of it. Fortunately for the team, some tools provide a server interface that allows for sharing and managing large documents - this was particularly prominent with the engineers working on the mechanical design of the rover who utilized SolidWorks. However, even this solution was specific to a single tool and not applicable across teams or sites.

Another key feature requirement of a document repository highlighted by the team is that it should be aware of both the formal and ad-hoc groups. This group awareness was desired to fulfill two different requirements: it would allow a member of a group to quickly browse all documents created by their group and find documents that may be relevant to their work, and secondly for the case of sensitive technology, it could be used to restrict access to only certain team members. In the context of space travel, this is particular important as many components and technologies may be under ITAR control by the United States government and require mandatory access controls.

A final feature that was requested by the team is versioning of documents. While the software team has long had automatic versioning of documents thanks to their use of Subversion, most other teams had no way to version their documents. Versioning was seen as addressing the need to backtrack to previous versions should a design change require it, and also allowed engineers to perform a freeze on a component that may require some integration and refer to the exact version they were working with - similar to what is done with integration testing in software.

These requirements are one area that it is particularly important for the team to get correct early as migration to a new document repository system is non-trivial. However, an examination of the market of available tools found no solutions that met the requirements proposed by the team, leaving the team with constant knowledge that the system will change in the near future. The most common issue was that systems were entirely web based, which is not practical for large files, followed by the inability to easily utilize and create static and ad hoc groups.

Presentation and Meeting Support

For many meetings in this early stage of team development, team members believed that the current system of emailing out slides before the meeting and having remote participants call into a conference line was sufficient to convey information. As a simple fallback, they believed that this could last for an extended time without radical changes. However, this structure relies on only having a few team members dialing into the meeting and requires that their participation is fairly low -- which is how the current "All Hands" meetings function. Once the team evolves and begins doing more design reviews and integration with the engineers at LPL and Raytheon, it is unlikely that this strategy will be successful.

The team identified several key technologies they believed could benefit the team as they grow. At a basic level, providing a tool to synchronize displays between local and remote participants would be beneficial for keeping meetings progressing without remote individuals being unaware that discussion had progressed to the next topic. Control of such basic screen sharing need not be two way for general meetings, especially if the meetings consist primarily of presenting information from slide decks. However, there are cases when it may be beneficial for individuals at remote locations to markup a shared document, or even interact with the remote display - for example when examining a map of the thermal loads placed on components of the robot. The team was aware of a variety of tools that could support these interactions, both through straight software solutions such as remote desktop, and integrated hardware/software solutions such as those from Smart Technologies, but declined to implement them right now. Their primary reason for not implementing these technologies was because there hadn't yet been a compelling reason, even though our observations showed there were clearly times when remote participants could benefit.

Another desire of the team was archiving of meetings for future reference and also for use in creation of promotional materials related to the project. This team currently has the benefit of our observations of meetings, which involve both video and audio recording of meetings. This process is resource intensive and requires an additional at each meeting to record, annotate, and then upload the video to a shared system where it can be later accessed and examined and is not possible for meetings that take place in Arizona. Several commercial offerings claim to support such recording and archiving, however the way in which information about the products is presented makes it impossible to evaluate the products objectively before making a purchase.

A final desire of the team was a tool that supports ad-hoc interaction between sites with video and voice support. Desktop users already have the benefit of video chat systems such as Skype and iChat, but they fail to scale beyond small group meetings and make it difficult to discuss physical objects. In addition, the need to take an action to start the video interaction, along with the potential headaches that can occur when starting such an interaction may deter casual users from adopting the system. Rather the team believed that it would be beneficial to have an audio/video link between the two primary sites that was always on. Systems that support such interactions have been mainstays of CSCW related research for many years [4, 5], however they haven't yet been widely adopted.

Complete Integration

One of the biggest problems with many solutions is that they provide little integration with common identity and group management infrastructure. Our preliminary analysis showed that team members may have as many as 8 different accounts and passwords to manage. When we asked a project manager if he believed that number was accurate his response was "I know it's at least that many." Given the difficulty of remembering usernames and passwords for each site, the team needs a way to manage all participation with a consistent identity. While each of the universities in the project both had the infrastructure present to manage this issue, there is no easily feasible way to link up the two identity management systems or provide such a system for team members not affiliated with one of the universities.

While there are some solutions to this problem, the primary problem is that there is not a standard that integrates well enough with a wide variety of tools. Many web tools have begun to adopt OpenID as a solution for identity management because it allows a federated infrastructure, exactly what this team needs. However, OpenID has yet to achieve widespread adoption even on the web, and very few non-web tools and commercial tools support OpenID. Many smaller teams utilize Google Apps for Domains, which provides most common services for a small team with a common identity framework, but, these accounts are not usable outside of the Google Apps environment. Most commonly it appeared as though the team would have to commit entirely to a single vendor to obtain this level of integration -- something that due to the other tool requirements seems unlikely.

Technical Expertise Issues

When starting a team, especially one that is adding members in an ad-hoc basis, the individuals who initially deploy the servers for collaboration technologies are often skilled engineers who were brought onto the project because of their skills in other areas of the project, and this project is no different. The proliferation of servers set up by different sub-teams within the project is evidence of this in action. However, as the team is growing, individuals who set up these servers have been forced to spend more time to customize the servers and software to the needs of the team and maintain accounts for members to access the data stored on the machines. Furthermore, the team believed the amount of work required to maintain the servers was highest around critical development periods, such as integration or near large scale project reviews -- exactly the moments that individuals have the least amount of time to maintain the servers.

While it was believed that as the team grew, they would eventually be able to hire a set of full time system administrators to deploy and manage the collaboration technologies, for the developing team this is a key issue and a large reason why the team initially chose hosted environments such as BaseCamp for many of their collaboration solutions. While hosted solutions reduce much of the workload, the knowledge that eventually the team will need to move away from these solutions as they require more customized tools means that there must be a way for the team to utilize data export functions of hosted tools to migrate all data to new hosted systems. In that sense, the team believed that solutions that supported standard data interchange formats would be most likely to support long term success of the team.

Cross Platform Functionality

A final issue making tool and platform adoption more difficult is that individual teams within the project have different desktop computing platforms that they choose to work on -- which are often directly related to the availability of tools for that platform. The software team tends to use tools that require either Linux or UNIX -- so most members use Linux or Mac OS X as their desktop operating system. In contrast, the tools for mechanical design of the project run on Windows and most of the team members who use those tools use Windows as their desktop operating system. On a day to day basis, this usually does not pose major problems -- our observations show that the Linux users are technically savvy users who can work solutions to computability problems on their own, and that for the most part compatibility is good enough between Windows and Mac OS X users, although on occasion issues still arise. For example, slides for the weekly meeting are often corrupted, either with misplaced elements or completely missing graphics, when the slides are transferred from one platform to another.

The heavy use of videos, both in the testing and evaluation of the project, and the requirement that the final project transmit video both for archival purposes and for driving the lunar rover introduces a new issue with cross platform compatibility. The team needs to select a video codec that is supported by the on board computer of the rover, the computers running the mission, and end user systems attempting to make use of the video. For the moment the team has avoided most of this issue by selecting Video LAN Client (VLC), an open source project that supports most video formats and runs on most platforms, as their player of choice. However, most video editing is done on Macs, which often produce files that VLC cannot play. In the case that the person supplying the video usually switches hooks their personal laptop to the video projector and shows the video through their personal computer. However, such a situation has significant problems for remote team members and data archiving.

The team has faced additional functionality issues when dealing with web based tools. Although the presence of Safari and Firefox has helped drive web applications to be more standards compliant, there are still older tools that required Internet Explore to function. In particular, a project defect reporting software which was popular with a team engineer who had previously worked at an aerospace corporation required Internet Explorer to report and view defects. Shortly after the system was set up the team member responsible was wondering why he never received any defect reports from testing - the reason is because all testing was done on Linux machines, making it difficult to report defects using Internet Explorer. In the end, the team member responsible for the system allowed individuals to submit reports to him via email, which he manually entered into the defect tracking system - effectively doubling the amount of time required to enter defects.

INTERVENTIONS FOR EXISTING TEAM

Sadly, given the requirements of the team and the available products, there is no simple magic bullet solution for the team that would address all of their needs. In particular, it appears as though most products will require a significant amount of customization that the team for which the team does not currently have the skills and resources necessary to perform. Even with significant customization, many of the commercially available groupware solutions such as Microsoft SharePoint, Lotus Notes, and Google Apps for Domains, have such significant drawbacks that they would not serve the team well.

One spot where CSCW related technology has made inroads to being publicly available is the field of real time video conferencing and collaboration. We are encouraging the team to utilize always on video conferencing systems between the primary sites at Carnegie Mellon and the University of Arizona to facilitate ad-hoc collaborations. Furthermore, we see great potential for real time screen sharing software between local and remote meeting participants and are working with the team to deploy such technologies.

In addition to recommending additional technologies for the team, we also recommended that there were some existing tools that the team should consider abandoning. In particular, we made the recommendation that the team consolidate mailing lists down to a manageable level. During our first examination there were more than 30 different mailing lists, some addressing very specific issues, most of which were never used. By consolidating down to a smaller number, we believe that users would be more apt to know where to send mail. Secondly, we also encouraged the team to drop their current ad-hoc method of scheduling meetings, which often means that many meetings go unknown. Rather we suggested that for the time being the team utilizes the calendar provided by Google Apps for Domains as it has excellent integration with other tools that many team members were already using.

We recognize that these solutions are only temporary, at best, but given the difficulties in implementing most solutions, we believed that this would provide a solid starting point for the team. Our next phase of research is to observe how these technologies change the interactions between individuals within and across sites as the team develops and matures.

RECOMMENDATIONS FOR FUTURE RESEARCH AND TOOL DEVELOPMENT

Clearly, one of the major issues that we faced in identifying and recommending solutions for the team was lack of commercially available technologies for collaboration. However, even if much of the technology from CSCW research was commercially available, it still would face an uphill battle for adoption because of the lack of integration between tools. Across almost all of the requirements, the team sought to have an integrated environment with consistent identities and group memberships. At one point during our requirements solicitation from the team, a project member illustrated a grand vision for integration of tools -- the ability to click on someone's name in an email and immediately call or message them, pull up files they've created, see what groups they're in, and understand how they relate to your work. In the case of a developing team, such knowledge is immensely beneficial because it allows members to easily contact each other and strengthen the bonds that are necessary for strong team formation.

In addition, we found that many tools were scaled for environments that were too big or too small for the developing team. Their current document repository solution works well right now, but as the team grows they'll need something bigger and more complex -- however most of these solutions do not perform well for small teams. This chasm between small team and enterprise scale tools means that most teams will need to make a dramatic shift during their lifetime; a shift that results in lost time and productivity and the need to establish new working norms. When developing solutions for team collaboration researchers should take into account the evolution of a team from small to large and attempt to provide solutions that grow with the team, promoting not only adoption of the tool, but also success of the team.

REFERENCES

1. Armbrecht, F. M. Ross et. al. 2001. Knowledge Management in Research and Development. Research-Technology Management, 44:4, 28-48.
2. Bradner, Erik and Gloria Mark. Why distance matters: effects on cooperation, persuasion and deception. In CSCW 2002. 226-235.
3. Cataldo, Marcelo et. al. 2006. Identification of coordination requirements: implications for the Design of collaboration and awareness tools. In proceedings of CSCW 2006. 353-362.
4. Dourish, Paul and Sara Bly. 1992. Portholes: Supporting Awareness in a Distributed Work Group. In proceedings of CHI 92, 541-547.
5. Fish, Robert S. et. al. 1993. Video as a Technology for Informal Communication, Communications of the ACM, 36, 48-61.

6. Halverson, Christine A. et. al. 2002. Behind the help desk: evolution of a knowledge management system in a large organization. In CSCW 2004. 304-313.
7. Handel, Mark and James Herbsleb. 2002. What is chat doing in the workplace? In proceedings of CSCW 2002. 1-10.
8. Herbsleb, James D. & Mockus, Audris. 2003. An empirical study of speed and communication in globally-distributed software development. IEEE Transactions on Software Engineering, 29:3, 1-14.
9. Isaacs, Ellen and John Tang. 1993. What video can and can't do for collaboration: a case study. In Proceedings of the First ACM International Conference on Multimedia. 199-206.
10. Isaacs, Ellen et. al. 2002. The character, functions, and styles of instant messaging in the workplace. In CSCW 2002. 11-20.
11. Nguyen, David and John Canny. 2007. Multiview: improving trust in group video conferencing through spatial faithfulness. In CHI 2007. 1465-1474.
12. Kraut, Robert et. al. 1990. Patterns of contact and communication in scientific research collaboration in Intellectual Teamwork: Social and Technological Foundations of Cooperative Work. Erlbaum Associates, Hillsdale, NJ, 149-171.
13. Olson, Gary M. and Judy S. Olson. 2000. Distance Matters. Human-Computer Interaction, 15, 2 & 3, 139-178.
14. Schmidt, Kjeld and Liam Bannon. 1992. Talking CSCW Seriously: Supporting Articulation Work. Computer Supported Cooperative Work. 1. 7-40.
15. Tuckman, Bruce. 1965. Developmental sequence in small groups. Psychological bulletin, 63, 384-399.
16. Tulio, Joe et. al. 2002. Augmenting shared personal calendars. In Proceedings of the 15th annual ACM symposium on User interface software and technology. 11-20.
17. Wenger, Etienne et. al. 2002. Cultivating Communities of Practice: A Guide to Managing Knowledge. Harvard Business Press, Cambridge, MA.