ABSTRACT

Sharing knowledge in a timely fashion is important in distributed software development. However, because experts are difficult to locate, developers tend to broadcast information to find the right people, which leads to overload and communication breakdowns. We study the context in which experts are included in an email discussion so that team members can identify experts sooner. In this paper, we conduct a case study examining why people emerge in discussions by examining email within a distributed team. We find that people emerge in the following four situations: when a crisis occurs, when they respond to explicit requests, when they are forwarded in announcements, and when discussants follow up on a previous event such as a meeting. We observe that emergent people respond not only to situations where developers are seeking expertise, but also to execute routine tasks. Our findings have implications for expertise seeking and knowledge management processes.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Management—Programming teams

General Terms
Human factors, Management

Keywords
Expertise seeking, collaborative software engineering, human factors in software engineering

1. INTRODUCTION

In global software development teams, team members have difficulty identifying experts and seeking knowledge from remote sites [4]. Developers make up for this by regularly communicating [3], but this leads to a tendency to broadcast email messages, causing information overload and leading to communication breakdowns [1]. Consequently, it is important to include the right people in an initial email. In our previous work [1], we observed that team members emerged as knowledge sources during the course of an email discussion—these people were not initially contacted, and were included in the email discussion only after a number of initial messages had been sent. Knowing who emergent people are and what they talk about leads us to better understand the process of expert knowledge seeking.

Developers acquire awareness of others [3] and frequently seek expertise from colleagues even if they belong to other projects or teams [2]. However, the authors know of no existing research that examines these emergent sources of knowledge. Why are emergent people included in these discussions? What knowledge do they contribute back to the discussion once they are included?

We examine emergent people in email discussion threads in a case study of a project within a large, distributed multinational corporation. We examine the contexts and reasons around which an emergent person shares knowledge with others.

2. RESEARCH QUESTIONS

A message thread is a series of email messages about the same topic. A message is sent to a number of initial recipients because the message is relevant to these people’s work. An emergent person is someone included in the thread who was not in the recipients list of the first email message sent in the thread. This person is included in the discussion when an initial participant includes the emergent person using carbon copy (CC) or forward (FW). Similarly, an emergent replier is an emergent person that contributes to the discussion by posting a message to the thread.

Our research questions are:

- What are the characteristics of discussions that include an emergent person?
- What situations lead to an emergent person being included in an email discussion?
- Why does an emergent replier decide to reply?
3. METHODOLOGY

We conducted a case study of a project team in a large corporation. The project team maintained an internal software product used by the company to support its shipping process. The product at the time of study was approximately seven years old, and was a critical component of the company’s business. There were two development sites in this project: USA, and Brazil. The team also communicated with production facilities located in other countries such as Malaysia. The team members work closely with each other, as well as with individuals outside of their immediate engineering team.

The development team was concerned with feature enhancements and provided support to internal clients and external clients. In addition to developing new features and providing support, the team was also responsible for the ongoing health of the system in production.

The project was delivered on time and on budget, with a small number of features moved out of scope. Based on interviews with the team members, the USA team and the Brazil team worked very well together, and they were proud about the quality of their product and their team.

3.1 Data Collected

We collected email inboxes from five team members in Brazil. These individuals were the test leader, the development leader, a senior developer, and two contractors performing development tasks. These team members communicate frequently with team members located in USA, India, and Malaysia, among other locations. The emails were sent over a period of eight months. We stripped attachments, and examined the messages for inline quoted messages. If the email contained an inline quoted message, we extracted it and saved the email as a separate message. This allowed us to examine email messages that were sent between members of the organization outside of these five team members. After conditioning, we had 4105 email messages.

We computed the number of emergent people in email threads, where a thread is identified by aligning emails that have identical subject lines. In this organization we found that few members tampered the subject lines. After arranging emails into threads by subject, we had 1095 threads.

Because we examined a corporate environment, email identities were relatively consistent. There were small variations, for example “John_Doe@example.com” and “Doe, John”, that were manually resolved.

We confirmed with the team members that email is the only primary method of communication used within the team. For them, written communication was important because of their distribution. In addition to email, the team used face-to-face meetings, teleconferences, instant messages, and an issue-tracking system.

3.2 Data Analysis

We selected the top 100 threads that contained the largest number of emergent incidents, where an emergent person replied to a message in a thread. We read these messages and used thematic qualitative coding to examine the common themes of discussion among the individuals, paying particular attention to when a person emerged in discussion, and when an emergent person contributed to the discussion.

After coming up with an initial set of codes, we further selected at random additional threads and re-coded the data to verify that the codes were in agreement with the data.

4. RESULTS

Email was sent or received by a total of 900 people. On average, an email thread had about 2 senders, 9 receivers, and 10 unique involved individuals, though some threads had up to 12 senders or up to 96 receivers. An email thread contained on average 3.8 messages, and ranged from 1 to 76 messages. The number of recipients and the number of emails per thread followed a power-law distribution.

Of these 1095 threads, 31% contained emergent people. When there is an emergent person, over 4 people on average emerge during the course of the discussion. The number of messages in a thread containing no emergent people is 2.2 messages/thread, but the number of messages in a thread with emergent people is 7.2 messages/thread.

Of threads with emergent people, 57% threads contain an instance where at least one emergent person replies to the discussion. There was an average of 1.7 emergent repliers. This means in most cases, someone included in a discussion often had something to contribute. Multiple roles were emergent people, including managers, developers, testers, and environment co-ordinators who manage servers.

We found that team members rarely removed quoted material from their emails; the entire thread of conversation was usually left intact. Repliers usually placed their reply at the top of a message, above the quoted material.

4.1 Contexts involving emergent people

We observed the following contexts in which an emergent person appears in an email discussion.

Crisis Situations.

As the system was a mission-critical system that was crucial to the operation of the business, it was often subjected to pressures that were outside of the organization’s control, such as infrastructure upgrades, manufacturing stoppages, and client requests. These unexpected events often affected large numbers of individuals, both within the team that we studied and outside of the team. In many cases, these unexpected events were crisis situations that had to be reacted to quickly. Often, a manager or senior team member was included in the initial message, and the rest of the technical team emerged as they were notified of the crisis. In many cases, the leaders requested information about how to properly plan for and implement the changes, or delegated the task to an individual on the team.

One example of such a situation is when major operating system patches were released. These patches were viewed as crucial and a corporate-wide message indicated that they were to be deployed during the weekend, less than a week after the initial notification. In this situation, the test team members were the emergent people. The development leader notified the test leader midway through the conversation, who had to come up with a test strategy of the system once the patches were put into place.

Another example occurred when a development leader was unable to locate a development server. The machine was not responding to pings, and was not in the physical location where it should have been. This issue was a crisis because the server was to be used for user-acceptance testing with a customer who was coming on-site. The emergent people were the development team members; the senior develop-
ment leader emailed the team to try to locate the machine, and later asked who was using which servers to see which machines could be reallocated.

In crisis situations, the message threads tended to be very long. Many emergent people were notified because crisis situations tend to “snowball”, meaning that new recipients are added. However, these threads did not involve many emergent repliers, suggesting that even though there were many people aware of the crisis, very few people were actually able to provide information to solve it.

**Explicit Requests.**

An emergent person often emerged as a result of an explicit request from an existing member in the discussion. A developer explicitly included a third party during a discussion to request that the developer investigate a specific issue. Even though this emergent person was not included initially, something that occurred during the discussion triggered one of the participants to request that this person perform a task. Related to this is expertise-seeking: a developer contacts another one to ask for assistance with a technical issue. If the initial recipient is unable to answer, then he will usually refer the sender to someone who does.

In this project, developers and testers did not always seek knowledge from others, but often requested others execute actions that did not require expert thinking. For instance, the test leader often required the environment co-ordinator to make changes to the test or production environments that the test leader did not have permission to make. Because of the critical nature of this project, access to various components was limited. The environment co-ordinators were often able to make the necessary changes in a timely fashion, but by forcing the team to go through these experts, the environment co-ordinators were fully aware of every change that occurs in the environment. Other requests we observed included software installations and execution of test scripts.

**Announcements.**

Emergent people were often included when an announcement was made. Such announcements were usually wide-reaching, and were sent by an upper-level manager. These messages were messages of congratulations or meeting invitations, but occasionally contained pertinent project scheduling information.

Because announcements were meant to address a large number of individuals, the developers in the project did not hesitate to forward the messages to individuals that they identified as not being on the recipients list. In one important situation, the new project manager sent a message that ordered the postponement of a database upgrade that was to happen this weekend. A developer who had been working for the project for approximately three years noticed that a number of contractors who were working on this upgrade were missing from the recipients list—this developer not only forwarded this message to the missing contractors, but also emailed the manager and informed him to notify the contractors about related issues in the future.

### 4.1.1 Following-up

Emergent people were often identified as a consequence of following up on an event, which was usually a meeting. The initial message, usually automatically-generated, was sent to a number of individuals. However, we observed in a number of cases that the message sent in reply to the meeting invite after the meeting was finished included a number of individuals who were not invited to the meeting.

We suspect that the emergent people were included for a number of reasons. First, they may have been invited to the meeting through word of mouth, not email. Second, the people who were included in the reply may be affected by the decisions made at the meeting, and the inclusion of these individuals was a matter of courtesy by the meeting host.

### 4.2 Discussion Topics Involving Emergent People

Project team members talked about many topics, including requirements clarifications, testing issues, issues with the environment, scheduling, project resource allocation, implementation, and I.T. support issues.

Technical discussions took place in email frequently, but surprisingly, the developers rarely discussed code. Instead, they frequently asked higher-level questions about how data is used, and what it is used for.

Team members, including developers, talked frequently with internal representatives of the business unit of the company to gather and clarify system requirements. The representative was not a marketing person and was familiar with how the system worked from a top-level perspective. He was able to report in detail on technical changes that were required in the system.

The team members also often contacted each other regarding coordination issues, but rather than coordinating activities such as code check-ins, they co-ordinated the exchange of information with teams who were interdependent with the shipping system. The teams exchanged artifacts such as test orders and manifest records.

One particular pattern we observed was a team member requesting permission to perform actions or requesting another team member to execute actions on his or her behalf. This organization, due to its mission-critical nature, appeared to employ strict access controls that individuals needed to work around. Developers and testers interacted frequently with owners of other subsystems and environment co-ordinators who managed the servers. This type of coordination falls outside of what is traditionally viewed as expert knowledge sharing, and may need to be treated differently than expert knowledge sharing.

### 4.3 Why does an Emergent Person Reply?

Even though 4.8 people on average emerge in threads with emergent people, only 1.05 of these people on average send a reply. This means that replying is quite rare. We examine why an emergent person chooses to send a reply to the other recipients.

**Consulting with Experts** The emergent replier consulted with the team to request a recommendation on how to proceed. This was not limited to developers; in fact a manager often CCed their technical team for advice. This behaviour is interesting because an emergent person is often presumed to be someone the participants seek information from, and not someone who taps into the discussion to opportunistically gather information.

**Resource or Status Updating** The emergent replier consulted with the team to request an update of their current status, usually with respect to resource use. This applied not only to physical resources, such as computers, but also
task assignments within the team. This was most commonly seen during a crisis situation, where an individual sends a message to learn who is using a particular server or who owns a particular piece of code.

**Reporting Results** During the discussion, a participant asked an emergent person to execute a task. The emergent replier responded with the results. This occurred frequently due to the limited access issue in the team, where developers and testers do not necessarily have access to make changes to other components in the testing environment or in the live production environment.

**Providing expertise** An emergent replier responded to a request for help. An individual asked who can help with a particular issue, and was referred to the emergent replier. The person receiving the first email always replied to the original sender, and CC the third party.

**5. DISCUSSION**

This research is new because it investigates an aspect of knowledge sharing that has gone relatively unstudied. Emergent people are a phenomenon that we have observed previously in software development. In a study of a project team maintaining a critical system, we found that emergent people are included in discussions that are crisis situations, in response to requests, as part of announcements, and inside follow-ups. In addition, we found that the team members in this project not only exchange expertise information, but also make routine requests to support staff.

This organization fostered a culture of open knowledge exchange, even between managers and developers. In this project, developers exchanged information, and often consulted with each other about how to proceed with certain tasks. Managers often sought advice from their development team and test team, especially with respect to planning how to proceed. As this team was responsible for ensuring that a production system was up-and-running full time, it was of utmost importance that the managers were aware of possible implications of any actions that they undertook. The organizational culture within this project group encourages open and direct communication and a feeling of co-operation. The emergent people were not limited to any one role; people from developers to testers to project managers were emergent.

In a crisis situation, finding the right knowledge was crucial to solving the issues. A context that often required knowledge exchange in this project were situations that were “crisis situations”. These situations were easily identifiable because they included a large number of emergent people and deep message threads. However, these crisis situations usually did not include a large number of emergent repliers. In a crisis, communication is spread to as many relevant recipients as possible, so the one person with the right expertise can step in with a solution.

Seeking out emergent people required time and effort. Thirty-five percent of the threads sent included emergent people, and of these threads, 58% included an emergent replier. Threads that included emergent people are longer, on average, compared to threads that did not include emergent people (2.45 vs 7.47). These threads were longer because portions of the discussion were devoted to letting everyone know that an additional person was included in the discussion.

Team members often communicated without exchanging expert knowledge. Nakakoji, et al. [5] distinguished between expertise communication and coordination communication. Nakakoji stated that coordination communication includes activities such as managing code conflicts. In our observations, many situations were simple and did not require expert knowledge. The most concrete example in our observations is the case where the test leader asks the environment co-ordinator to turn off a server so she can continue with her work. This form of coordination does not fit into the traditional definition of knowledge sharing, because the actions that the recipient carries out for the sender is a relatively routine task.

**6. IMPLICATIONS**

This work contributes to the field of knowledge sharing and expertise seeking. This study of emergent people gives us insight into how people communicate. We learned that a culture of open knowledge exchange was nurtured, and that not all communication is expert knowledge exchange. A culture of knowledge-sharing helps create opportunities for successful expertise-seeking. This co-operation led to the team’s success in delivering on time and on budget.

This work also emphasizes the importance of good expertise seeking. By reducing the number of threads in which an emergent person must be included, we may be able to shorten the size of communication paths and reduce the time required to resolve issues.

Awareness of emergent people may also form requirements for better knowledge management policies. In our organization, we observed that team members did not modify subject lines or remove quoted material, and followed a specific practice of CCing. Keeping this information intact benefits emergent people in the case that they do emerge. Some knowledge-seeking behaviour, such as following-up, may benefit from tool support, but others, such as crisis situations or resource or status updating, require timely communication instead.

We plan to continue our qualitative analysis to further develop out qualitative findings, and intend to augment it with quantitative methods such as social network analysis.

**7. REFERENCES**


