Studying Reactive, Risky, Complex, Long-Spanning, and Collaborative Work: The Case of IT Service Delivery

Eser Kandogan, Eben M. Haber, John H. Bailey, and Paul P. Maglio

IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120
{eser,ehaber,baileyj,pmaglio}@us.ibm.com

Abstract. IT service delivery is challenging to study. It is characterized by interacting systems of technology, people, and organizations. The work is sometimes reactive, sometimes carefully planned, often risky, and always complex and collaborative. In this paper we describe how we've learned about IT work, using a variety of methods including naturalistic observations, contextual interviews, surveys, and diary studies. We provide examples of our study results, showing what we've learned with the different methods. We argue that to effectively study such systems, a variety of methods may be needed to complement insights and validate findings. We found that naturalistic observations were extremely time and labor intensive, yet offered us the time and space to observe unplanned events and long-lasting tasks, bringing out the full complexity and risks involved in real work. Contextual interviews and diary studies provided fewer details, yet gave a broader context to individual's work. Surveys provided an even broader picture, going beyond individual differences, yet they were limited by details and issues of sampling.

1 Introduction

IT systems change constantly under long pressures from business, regulations, globalization, and technological change [5]. Over the short-term, system workloads and configurations change rapidly in response to day-to-day demand fluctuations. System administrators must balance both long and short-term needs: when errors and problems occur they must react to them as quickly as possible, yet the risk of failure demands careful long term planning for future growth and deployments. The work is often dynamic and reactive, yet the large, complex systems require long processes and substantial collaboration to configure and operate. For example, reorganizing databases may take couple of days or weeks and involve a number of specialists. To cope with this environment, practitioners have developed a wide variety of tools, methods, and organizations to effectively deliver IT services.

We have been studying IT Service delivery since 2002 to understand human issues in service delivery, particularly to inform the design of next generation systems [1]. Given reactive, risky, complex, long-spanning, and collaborative nature of the work, it is a challenge to study IT Service delivery - technology, people, and organizations introduce multiple interdependent variables. Short user studies, as typically conducted in usability labs, simply cannot accommodate the full complexity of real-world

J.A. Jacko (Ed.): Human-Computer Interaction, Part IV, HCII 2009, LNCS 5613, pp. 504–513, 2009.

administration work, and they ignore the everyday constraints and issues that make the work dynamic and reactive to the unplanned events. We believe short lab studies offer only limited insight into use cases and fail to sufficiently explore the design space.

We found ethnography a good fit for studying IT administration work. Naturalistic observations and contextual interviews conducted in the field allow inquiry of phenomenon within its real-life context, allowing time and space to see unplanned, transient and long-spanning events and tasks, all of which we found to be characteristic of system administration work. Field studies allow the observer to be situated in the work environment for extended periods, and they help facilitate in-depth study of the development, adoption, and use of new tools, practices, and organizations. We complemented field studies with diary studies and surveys to validate our findings across broader time scales and populations, though we achieved less detail with these methods. In this paper, we present examples from our findings to demonstrate the benefits and challenges in our study methods.

2 Studies of IT Services

We conducted a series of studies in IT service delivery organizations employing methods including naturalistic observations, contextual interviews, a survey, and a diary study. In 16 site visits to large corporate, university, and government data centers across the United States, we observed and interviewed more than 30 IT workers. We observed the work practices people involved in management of security, databases, web sites, storage, operating systems and data center operations [1]. Our methods were as follows:

- Contextual Interviews: 30-90 minute interviews with subjects at their desks, inquiring about their work and encouraging them to show us examples of important tools, techniques, and artifacts they use on a daily basis.
- Naturalistic Observations: following, observing, and if possible recording all the work-related activities of an administrator for anywhere from a few hours to a week. This usually involved two researchers, one to operate audio/video recorders, and the other to take notes and occasionally ask questions (though trying to avoid interrupting the flow of work). We asked participants to speak aloud if possible, and to put phones on speaker. At the end of each day, we asked clarifying questions about the observations from that day. Additionally, we collected physical and electronic materials and took pictures their work environment.
- Surveys. To validate our findings across a wider population, we administered one survey on tool use to 100 administrators. We have also worked with the SAGE organization which performs an annual survey on 1000-10,000 sysadmins, looking through their data for further correlations, and also adding a few new questions.
- Diary Study. One of our observation subjects was willing to share his 10-month diary listing, for each day, all the work-related activities he had performed that day.

Below we present samples from our findings to illustrate the results from each method. The first case shows how naturalistic observations were able to illuminate the extremely complex world of security administrators at a university computing center. In the second case, we employed contextual interviews to examine an organization

delivering storage management: Centralized Storage Services (CSS[1]). Here, contextual interviews helped us gain insight into the workings of the larger organization through descriptions of work processes that could take as long as a month with multiple people across different organizations. In the third case we describe our findings from our survey of trust issues in the design of system administration tools [6], which helped validate and refine some of our observational findings. We also describe some of our experience working with the data from the SAGE survey. Finally, we discuss a diary study that provided long-term validation for our observational findings with respect to collaboration between administrators.

Let's describe each in more detail.

2.1 Case 1: A Game of Cat and Mouse…

We conducted two separate week-long observations of the five-member data center security team at a large public university in the United States. We saw how their day-to-day work included continual monitoring for new security threats, policy formation, and helping remove security vulnerabilities. We also got to watch a notable episode, as they responded to a world-wide security incident, Stakkato. Stakkato had persistently attacked military, educational, and government sites across the United States and Europe. We observed the senior security administrator, Joe, handle this incident through both local work and also collaboration with other educational and government institutions. Collaboration was necessary because the attacks came through a complicated maze of computers in as many as 7 countries, making tracking very difficult. Joe was working with Aaron, a junior security administrator, and others to respond to the attacks. This was very much a game of cat-and-mouse: the attacker would use vulnerabilities to gain access to machines, while the security admins knew about vulnerabilities in the attacker's tools that they would use to try and trace the attack back to its source. Often a compromised machine would be left vulnerable to allow more time to trace the attacker.

While the center had not been attacked for couple of weeks, the attacks remained a primary concern for Joe, and appeared to have become a personal issue for him. Thus Joe seemed sad and frustrated when he received the news that a compromised machine at another institution was turned off – he wanted the machine to stay on so he could continue to use it to trace the attacks, yet the institution that owned the machine didn't want to leave it vulnerable any longer.

Tracing attackers was no easy task. It was technically challenging as security admins needed to exploit the particular techniques the hackers were using. On a social level, it was even more challenging as they needed to coordinate work with several sites and develop an understanding to share information that could be sensitive. Security staff at a number of institutions held weekly calls and exchanged information, tools, and strategies through email, phone, etc. Joe played an important role coordinating the effort and communicating their findings:

Joe: Usually they [the attackers] come in from Europe to a machine in the US, and they make either one or two more hops before they start launching any attacks. We had that narrowed down to where they are coming in, but that site wasn't able to help

[1] To protect the identity of the people, organizations, and companies, we used fictitious names throughout the paper.

us. So, we knew that machine, we knew the second machine they are hopping to, and there is a third machine that was here in town that they are using as well. I need to call them today, and I need to track down some of these other sites, the admins there.

The particularly challenging issue was that the attackers kept coming back relentlessly using new malware. Upon reviewing session logs Joe noticed a number of new exploits and asked Aaron's help to find more out about them. Joe and others at the center used an elaborate directory structure to collaborate on incidents. They had directories for each incident to keep scans and exploit code organized:

Joe: They have two new things in there, that I didn't notice. I told them about identity key there (pointing to ingresd.x.x.edu) and I am not sure what it is from. They are trying to figure out what it is from as well. […] But that mod_rootme, we need to find out what that is. […] The other thing they are using is that ussl thing. Those are two new tools, I haven't seen them used before.

Aaron's research on several security sites revealed that mod_rootme was a high-threat vulnerability of Apache Web Server. It allowed a remote user to get a root shell without being noticed as communication was disguised as normal web traffic. He also found a copy of the exploit source code and noticed a couple of identities there:

```
printf("[*] named 8.2.x ( < 8.2.3-REL) remote root exploit by lucysoft, Ixix");
printf("[*] fixed by ian@cypherpunks.ca and jwilkins@bitland.ne\n\e");
```

A web search on these identities, i.e. lucysoft, ian@cypherpunks.ca and jwilkins@bitland.ne, led him to the BIND exploit. Aaron explained that while it was very difficult to understand source code, references to identities often led him to further clues on the exploits:

Aaron: […] and that is what I generally look for. Like most of them have assembly code or something else. Rather than spending time on what exactly the exploit is doing, first I try to look at the comments, explanations, and signature of the authors. Because generally all the hackers they tend to write their own signature handle.

Aaron also noticed another source code (a.c) in the sessions which he discussed with Joe, later in his office:

Joe: Okay. Where was that a.c? Was that in there??

Aaron: Uhh. Yeah, it is in canopy.

Joe: So, this is a BIND one, huh? We should be able to build, put that together.

Aaron: main.c and main.h

Joe said they would often try these exploits in quarantine environments on virtual machines with limited connectivity and learn how they worked. When Aaron got back to this office and began working on other tasks, he got an email alert from the intrusion detection system, Bro. Aaron and his colleagues used several automated systems that scanned network traffic and computer activity for suspicious events. This alert was about new activity on a formerly-compromised host:

```
Watchdog found the following alerts in tcpread. These seem to be coming from
the known compromised hosts. Please take time to investigate.

Non-XX IP (Once Compromised IP's)
x.x.31.10 #nyx.engine.xx.edu (6/8 from victor, used as login to XX)

XX IP's Connections
> Jul 27 15:10:14 x.x.31.10 0.1kb > x.x.63.22/http 711kb 0.0b %77125
```

The alert stated that there had been recent network activity, an HTTP transfer of a file of size 711kb with the log ID 77125, to a local machine. This was a machine reported as compromised by Victor, a fellow security administrator at another site, on 6/8. Aaron had customized the intrusion detection system to register certain hosts as compromised and added further contextual information such as who reported it and when and how it was compromised in the past. This alert was of utmost importance as the host in question was directly related to the Stakkato incident.

When Aaron examined the HTTP logs he found out that a particular file had been downloaded. Aaron first identified the host name, and then searched for the owner of the machine on an online database. Aaron immediately did a search and found his home page. Examining web page he discovered that the user worked on high performance computing, making the file in question a legitimate download. Both Tom and Aaron were relieved as this potentially serious alert turned out to be a false alarm.

Responding to a security attack required a range of activities, from minute-by-minute monitoring of activities to long-term research, planning, and collaboration across multiple sites. Security administrators have a fascinating culture, always trying to learn about vulnerabilities of the enemy, while preventing the enemy from realizing what is known about them. Naturalistic observation was very helpful in enabling us to gain a detailed understanding of the full richness of this environment.

2.2 Case 2: Everybody Thinks That They Are Number One

We conducted a three-day study with several contextual interviews in the Centralized Storage Services (CSS) group at a large IT Service company. CSS offers enterprise-scale storage solutions centered at one location, with managed storage resources distributed over 10 sites across the world. With a novel offering that delivered on-demand storage capacity allocation, 24 by 7 monitoring, and management services, CSS grew its customer base from only 2 customers with 2 high-end "SHARK" storage systems to 27 customers with about 50 SHARKs, just within two years.

We interviewed Henry, who worked in the Storage Operations Center (SOC) as the storage team "focal" (i.e., project lead) since CSS's infancy. Henry's group handled anything having to do with storage, from technical work such as updating microcode software to coordinating work such as negotiating change windows with customers. Henry explained that customers got more or less equal treatment from CSS, though certain tasks took a higher priority:

Henry: Everybody thinks that they are number one, you know. So, our model is that nobody is given priority over anybody else. Yes, we set a priority over how it has to be done cause basically if there is an outage, lets say, filesystem has reached 99.99%, and it is about to crap out and cause the server to break down, we may have to push doing that allocation before that one.

Nonetheless, with rapid customer growth they hit technical limits that required significant human intervention, as illustrated by one incident the previous week:

Henry: The SHARK has a limitation that allows you so many port logins. We hit that limit. So, support came back and said that the only fix for that was to recycle [reboot] the host bays. So, now you gotta get approvals from the customer. That is a problem. We had like 50-60 servers, which has a different application for each one. You gotta get

approval from all these people. So, we finally got approval. Problem was fixed but it took like a week and a half to two weeks away, just to get approvals. Customer is happy now. They see the storage, but we hit one of those limits that is in a book like that thick.

Rapid growth not only brought technical limits but also organizational issues to deal with. A simple setting, maximum number of port logins, when exceeded required approvals from each customer. While customers were isolated from each other in terms of service quality, we saw that collective use of shared storage resources led to new issues that would not have risen otherwise.

In CSS, they aimed for no outages and downtimes. This meant that operations needed to be performed concurrently, i.e. without taking the whole cluster down. One of the critical issues in concurrent operations was microcode updates. To address this issue Henry and others developed a 4-week process, depicting tasks and interaction with customer as a flowchart. The first two weeks were primarily for aligning requirements and schedules with the customer. Handshaking occurred from both a technical perspective and social perspective:

Henry: It is a four week process from the point we approach the customer. We establish the [change] window, we tell them here are the servers we see you have, storage on your SHARK, and levels of SDDs you running on these servers. A lot of times what you see on the SHARK is not what you get. So, you have to make sure there is a handshake between the zoning and the SHARK. So, between weeks 2 and 3 we do all that handshaking making sure that SAs are aware. Is it compliant, if not we tell you, hey are you willing to bring down your server?

The remaining two weeks were primarily for performing technical work. Week 3 was for upgrades to meet minimum requirements. Week 4 was final check with the customer detailing schedules, contact information and technical details to give a last chance to withdraw. Once SOC performed the change they closed the ticket with a final report to the customer.

Synchronization and coordination of work among parties always presented a challenge. Henry mentioned that when storage work was completed in SOC, it might still take time to bring up the servers to verify the change. And he found it always frustrating when customers could not see their storage and he had already moved to another issue. Remembering changes from weeks before was a burden, especially as administrators were not devoted a specific customer. To help with this they developed a database to keep track:

Henry: Do you really remember what you did four weeks ago? We have a database that can house that information so you can pull it up easily. Because if I am working on customer A today, three weeks from now, customer C says, well, you worked on my stuff a month ago. Customer A is on my mind right now. I am not thinking about customer C!

Henry described how most of these challenges at CSS were manageable through careful planning and processes developed in the SOC. Henry and others developed spreadsheets and flowcharts to guide work and coordinate with customers. Lags within the organization presented a special problem. Differences in storage design, allocation, and server board times were particularly difficult to deal with. Solving these issues required invoking organizational memory to recall problems, participants, and solutions to revisit and resolve them with the right people and tools.

2.3 Survey Studies

In our field studies we observed numerous cases where the tools failed to support system administrators, reporting incorrect data, working unreliably, or simply being poorly aligned with the administrator's work practices. Given the risk factors in the IT administration we expected that trust played a major role in the way administrators used and interacted with information, people, tools, and systems. To see if these findings were valid across a broader population, we performed a survey of about 100 system administrators recruited through online news groups, and local and national system administrator organizations. In the survey we asked specific questions about administrators' comparative qualitative judgments of the CLIs and GUIs for system management they use regularly. One set of questions concerned the perceived speed, ease of use, reliability, robustness, accuracy, trustworthiness, and likeability of the CLIs vs. GUIs they used for system administration. Additional questions about both CLI and GUI tools were based on McAllister's survey, using a 7-point Likert scale, which measures monitoring behavior and cognition-based trust for interpersonal relationships in organizations. The details of the survey results are included in [6]. In sum, we found statistically significant preferences for command-line tools over graphical tools in the areas of trustworthiness, reliability, robustness, accuracy, and speed. In response to the McAllister questions, we found that cognition-based trust ultimately plays a major role as opposed to seemingly affective factors. We received some comments, such as, "I know what I am doing. Please NO MORE GUI! If people need a GUI they are not qualified to be doing whatever they are trying to do."

In search of further statistically grounded data, we also worked with SAGE, the System Administrators Guild, which annually administers a survey intended to collect demographic and salary information. This survey is very large, with between 680 and 9600 respondents depending on the year. The survey includes questions about supervisory roles, and also administrators per site, from which estimates of administrator team size can be inferred. Further, we were able to add some explicit questions about collaboration into the 2007 survey. The results suggest that administrators work in teams regardless of company size, and that the majority of them spend a third or more of their time working with others. We did need to carefully formulate the questions to make them easy to answer, since subjects often failed to answer complex or open-ended questions.

2.4 Diary Study

George was a web-application administrator in a large corporate service delivery center. We observed him in two separate week-long field studies conduct several tasks such as installing web servers, configuring authentication servers, trouble shooting problems, etc. For his own purposes, he kept a record of tasks performed daily, such as troubleshooting (e.g. "Continued with webseal problems on acp2"), and meetings (e.g. "RFS customer call - 16 new servers coming in"), etc. On a typical day, he had about five to ten records. He did not, however, attribute the amount of time spent on each task.

George was willing to share his diary with us. We analyzed it by categorizing tasks in each record, and by noting what tools and people were mentioned. Our purpose was to get a break down of tasks over an extended periods and get a sense of the

extent of collaboration in conducting tasks. Our analysis confirmed that collaboration was indeed practiced extensively with about 23% of the records referring to meetings with other system administrators. Of the remaining tasks such as planning (21%) and trouble-shooting (11%) , the diary mentioned other admins nearly half the time. Across all tasks he indicated collaborating with others about 45% of the time. This was a high number even though we suspected that collaboration could have been downplayed in self-reports. We were able to evaluate self-reporting by comparing a 3 hour troubleshooting session we video-recorded with the diary for the same day. In coding the session, we observed him working with one of ten other people for 90% of the time. His report, however, discounted the extensive collaboration simply saying, "Communication problem with PD server - we had port opened in wrong direction - were able to use extra port 7236 - works fine now." The only indication of teamwork was mentioning "We" as opposed to "I".

3 Analysis: Multiple Methods for a Complex Topic

System administration work is clearly a challenging area to study. The work is sometimes reactive, sometimes carefully planned, often risky, and always complex and collaborative. We have found that a variety of methods is necessary to generate a complete and accurate picture of administration work: ethnographic approaches such as naturalistic observation and contextual interviews, as well as surveys, and diary studies. Consider the case of the security administrators reacting to alerts about potentially suspicious activity. It is hard to imagine any way of capturing the intricate and coordinated response and evaluation of the alerts, short of watching them unfold.

Consider also the complexity of the technological arms race and information warfare going on between the security administrators and their adversaries. The admins are continually researching vulnerabilities in their own systems. They must also search for the often subtle signs of intrusion, and try to find vulnerabilities in the intruders' tools. Attacks, when detected, are often permitted to continue, in order to trace them back to their source. Any data left behind by the attackers must be analyzed, reverse engineered, or even decrypted to find clues about the attack's origin. Furthermore, attacks often come from compromised machines at other institutions, requiring collaboration across the broader community. Observation seems to be the only way to get a clear picture of this very complex environment.

Contextual interviews are an important complement to naturalistic observations, they provide the context, history, motivation, and descriptions of exceptional or rare events that might be difficult to tease out from observations alone. Our interviews with Henry about his storage administration work gave us insights into the history of his group, and the clearly defined workflow that formalized the interaction among the participants. For example, Henry explained the elaborate flowchart for the four-week long interactions with the customer when updating microcode software. A significant part of the interactions focused on synchronization and coordination. We learned that rapid growth in the customer base led to technical issues, which in turn led to a weeks-long process to get approvals from all the participants involved and perform changes. Dynamic growth also required them to perform flexible resource allocation. This reflected in the way CSS got organized over long periods of time.

An ethnographic approach was extremely valuable in studying IT service delivery because it allowed us to capture work broadly and flexibly in time and space with all the interdependent variables of technology, people, and organizations as they emerged during the course of study. Thus, we argue that to understand fundamental issues and formulate hypothesis of work in the context of larger systems of people and technology an ethnographic approach is invaluable. The most common alternative to ethnography, in-lab user studies, begin by eliminating or controlling variables, thus they are more suited to achieve shorter term goals such as measuring usability or task performance. Another alternative is modeling usage in a computational way in terms of actions and affordances. This approach does help to get statistical data on performance, yet it may be neglecting important aspects of experience [3, 4]. Ethnography goes beyond actions of individuals and considers cognition also from a social perspective. Thus, ethnographic studies are better suited for verifying hypothesis. Expecting similar short term goals such as design guidelines from ethnographic approaches is not the best criterion for the relevance, utility, and quality of an ethnographic account of work, and such an expectation indeed underplays potentially radical gains that can be achieved by laying out a broader design space [2].

Ethnography does have its limitations or challenges, too. First, ethnographic methods are expensive. It takes considerable time and commitment to conduct long-term observations. It also takes time to establish the trust of the members of the study group, who must put up with being observed, followed, and questioned. We found it considerably easier to obtain permission to observe and record employees within our own company; obtaining the legal agreements required to do such work at other companies proved to be extremely challenging. Once the data was collected, it took even more time to analyze: an hour of video might require a day to a week to truly understand. We even developed our own video analysis tools to speed up the process of video analysis and annotation. Ethnographic studies are not ideal for developing low-level design guidelines that necessitate quantification of user needs and performance. Absence of quantification, and lack of control for variables are often used to question the validity and reliability of results though both issues are addressable when ethnography is used to derive hypotheses. Validity is addressed by collecting data from diverse situations and sources and considering all to expand hypotheses instead of discarding any data that did not fit a hypothesis. Reliability is addressed by triangulation where data is collected from multiple sources, across situations, and time. Last but not least, ethnographic accounts are descriptive rather than prescriptive thus making multiple interpretations a possibility, a quality that can in fact be considered as an advantage.

We faced several challenges as we conducted our ethnographic field studies. First, the work of the system administrators was very complex, and outside of our own areas of expertise. It was nearly impossible for us to track and understand commands, errors, and discussions surrounding them as the work was unfolding. Likewise, a lot of the work was dynamic and concurrent, many issues were being addressed at the same time, through simultaneous conversations among staff and interactions with multiple systems. To address these issues we utilized video analysis to help us better understand the work. However, it still required extensive effort afterwards to make sense of all the technical issues and unfold multiple concurrent interactions into coherent parts, and establish cause and effect. All this required meticulous attention to every detail, on the computer screen and utterances of the participants, as well as

technical documentation studied after the fact. Tracking issues from multiple perspectives of the participants also helped to the extent possible.

Unlike traditional anthropologists, we were not able to spend months or years with our subjects. This meant that our observations and interviews drew from a limited sample. Data sources such as George's diary were extremely valuable in showing that collaboration practice was not an isolated case but occurred on longer time spans across different types of tasks.

The extremely labor-intensive nature of naturalistic observation also left us with a limited population sample. Surveys were one means to validate findings across a broader population, especially when we could frame very specific questions to ask.

4 Conclusion

The purpose of this article is to discuss the strengths and limitations of various study methods as applied to the dynamic, risky, complex, and long-spanning work environment of IT service delivery centers. We argue that observational methods offer both time and space to observe short-term reactive work and long-lasting phenomena and tasks with its full complexity with all considerations of the interplay of technology, people, and organization systems through detailed illustrations of two cases from our field work. We also describe how contextual interviews can provide further background, details, and historical information. We tried to address some of the shortcomings of ethnographic approaches through surveys and diary studies that offered validity beyond time and individual differences. Finally, we discussed challenges we faced, and made recommendations on possible ways to address them.

References

1. Barrett, R., Kandogan, E., Maglio, P.P., Haber, E., Takayama, L., Prabaker, M.: Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. In: Proc. of the CSCW 2004, pp. 388–395. ACM, New York (2004)
2. Dourish, P.: Responsibilities and implications: further thoughts on ethnography and design. In: Proceedings of the 2007 Conference on Designing For User Experiences (2007)
3. Norman, D.: Emotional Design: Why We Love (or Hate) Everyday Things. Basic Books (2004)
4. Picard, R.: Affective Computing. MIT Press, Cambridge (1997)
5. Spohrer, J., Riecken, D.: Introduction. Communications of the ACM 49(7), 30–32 (2006)
6. Takayama, L., Kandogan, E.: Trust as an underlying factor of system administrator interface choice. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 1391–1396. ACM, New York (2006)