Challenges, Motivations, and Success Factors in the Creation of Hurricane Katrina "Person Locator" Web Sites

Christopher Scaffidi[1], Brad Myers[2], Mary Shaw[3]

[1] Institute for Software Research, Intl., School of Computer Science,
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213

[2] Sloan Software Industry Center and
Institute for Software Research, Intl., School of Computer Science,
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213

[3] Human-Computer Interaction Institute, School of Computer Science,
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213

{cscaffid, mary.shaw, bam}@cs.cmu.edu

Abstract. We interviewed six people who led teams that created web sites enabling Hurricane Katrina survivors to report their status. We learned that interviewees did not discover and communicate with other teams when they started their projects, which resulted in redundant sites. The absence of a shared task impeded trust between teams, ultimately inhibiting data collection and aggregation. Moreover, communication within teams was problematic; developers who had adequate technical skills to work alone were more positive about their sites' success compared to developers who had to shore up skill weaknesses through collaboration. These problems did not simply result from team leaders' over-sized egos, since site creators were generally motivated by concern for other people instead of self-interested motivations. Rather, these problems highlight the need for improved development methods and systems to help developers discover and communicate with other teams' leaders in order to collaborate on widely distributed, time-critical projects.

1 Introduction

On Aug. 29, 2005, Hurricane Katrina made landfall near New Orleans, LA. The storm breached levees, cut off phone and electricity service, flooded homes, and displaced hundreds of thousands of people throughout the Gulf Coast.

In this tragedy's wake, people turned to the web to learn whether friends and family had survived and, if so, where they took shelter. To assist in this search, dozens of people led teams that created web sites so users could store and retrieve data related to survivors' locations. The decision to lead these teams required significant invest-

ments of time, emotions, and skills. These leaders' efforts proved instrumental in reassuring and reconnecting many people.

We interviewed creators of six Hurricane Katrina sites (HKS) to investigate two primary issues: What challenges did they encounter? What factors motivated their volunteerism and contributed to success? Because HKS creators operated under tighter time pressure than other volunteer software developers (for example, in the open source community), we anticipated that interviewees would provide unique answers to the questions above.

Section 2 summarizes our sample and method. Section 3 describes challenges to helping users locate people. Section 4 discusses interviewees' motivations, which differed somewhat from motivations of volunteer software developers in the open source community, while Section 5 analyzes success factors. Section 6 outlines opportunities for future research aimed at facilitating highly distributed, time-critical software development.

2 Sample and Method

Our research group's objective is to help "non-professional programmers" with minimal training effectively create software such as web applications. Therefore, rather than interviewing every HKS creator, we wished to focus on sites that appeared to have been constructed by volunteers on a tight budget without the sponsorship of a large corporation.

Consequently, of 22 sites located using search engines, we narrowed our interest to those that did not display any corporate sponsors' logo, had only a few pages, and lacked fancy graphics. Combining these criteria yielded 10 sites. In addition, we targeted one "aggregator" site that consolidated other sites' records; we included this site because a companion Wiki site displayed numerous postings from volunteers who had worked on this site. As discussed by Section 3, each site contained records for hundreds to thousands of hurricane survivors. We regularly browsed the 11 sites from September through December and screen-captured many pages on Nov. 29 in case if teams took their sites down.

We used DNS registration and information on the sites to determine each site owner's email address and phone number. Six HKS creators agreed to participate in one 30-minute semi-structured telephone interview each between Nov. 4 and Nov. 11. We tailored questions to address our primary focus: motivation and success factors.

We performed cross-case analysis by question to identify common concepts mentioned by respondents. Based on question-level commonalities, we coded responses into a matrix (respondents on rows and questions on columns) to identify patterns spanning questions. We reviewed preliminary findings and raw data with another researcher from outside our group to verify that the data adequately and consistently supported our findings.

3   Experiences of HKS Creators

HKS creators' first spur to action was a sense of chaos. One watched television, another read a blog, a third got a phone call, a fourth was unable to complete a phone call—they each realized that hundreds of thousands of people were fearing for one another and unsuccessfully searching for loved ones.

By Aug. 31, one interviewee had created a PHP-based site with web forms that visitors could use to store location information in a publicly viewable database. Over the next four days, two more PHP-based sites and one blog-based site came up.

At this point, interviewees were unaware of one another's sites; they simply took the tools that they knew—PHP or a blog—and built as fast as they could. They sought simplicity in design. For example, the blog lacked a web form for posting information; users had to email the blog's owner so he could post the information. Some interviewees did not plan, build, or host their site entirely on their own, but rather sought help.

These sites drew an overwhelming response. One site received over 9000 unique visitors per day for the first week; another had traffic exceeding 60 MB/second during the same period, tapering off over a month. Testimonials poured in through email and sites' "Contact Us" forms. HKS creators happily posted some notes online, such as "Thank You!!!!!!!!!!!!!!! Thanks for your site. … Out of 10 people, seven so far are fine, and found thanks to you!"

Sadly, not all notes to HKS creators were joyful. Many interviewees spent most of their waking hours answering hundreds of messages from site visitors asking about loved ones. HKS creators could tell them no more than the information already posted on the site, and the frustration and long hours gradually took an emotional toll. One person described the experience as "playing the role of counselor," while another "was in bad shape" and "wasn't well." He compared it to the creation of a similar site after September 11, which "also ate my life for a couple of weeks."

Three main challenges made it hard for HKS creators to help users find each other:
- Users had to search dozens of person locator sites.
- HKS creators had to fend off requests by distrusted aggregators for databases.
- Infrastructure damage impeded getting data out of hurricane-stricken areas.

Redundant sites: First, users had to search dozens of person locator sites, each of which had a unique user interface and database. Unfortunately, HKS creators had no way to know that other people were in the process of creating sites. Even search engines, the best existing tools for locating web sites, were of little use immediately after this disaster because they take time to "notice" new sites as they crawl the web. Interviewees who asked the media to advertise their sites were generally disappointed: in fact, one HKS creator mentioned that when he told radio and television stations about the site, they each opted to create an HKS of their own! (Fortunately, bloggers, newspapers, and government agencies advertised sites over the course of several weeks.)

Recognizing these inefficiencies, two interviewees collaborated with other people to build a site that aggregated data from many other sites; this site went live on Sept. 5. In addition, one of these two interviewees also contributed to a second site that

published an XML standard related to the aggregator site. The aggregator site was Java-based; the XML site was static HTML.

Whereas earlier person locator sites grew as visitors manually entered data, aggregators tried to populate their site automatically using screen scraper scripts that read data off older sites. Some source sites, particularly those based on free-text tools like blogs, were not amenable to this approach; in this case, aggregators browsed those sites and manually copied data items from old sites into the new site. (The new site also had a page where owners of older sites could upload XML containing data, though they did not use this page as heavily as anticipated.)

Distrusted aggregators: Three interviewees said the biggest challenge was fending off requests by aggregators for databases. They primarily resisted sharing with aggregators to protect their users' privacy. Not to be dissuaded, aggregators screen-scraped to acquire data. (Ironically, one of the three interviewees who complained about aggregators was himself an aggregator, and his team was eventually threatened with a lawsuit because they had aggregated data from a site without permission.)

One illustrative case was a non-profit organization that requested data from these three interviewees. In addition to a desire to protect their users' privacy, the HKS creators refused to share data because they perceived the organization's staff as unresponsive to questions, technically incompetent, and generally pushy. Moreover, interviewees felt that the organization's HKS was too late to be useful, not user-friendly, and lacking in search features. The HKS creators' resistance is understandable, since their interactions with this organization lacked the five ingredients necessary for trust in distributed virtual teams: enthusiastic social communication, competent and timely responses, individual initiative and leadership, focus on a shared task, and a cool reaction to crisis [1]. Without these ingredients, the relationships between HKS creators and the non-profit organization soon grew antagonistic.

Broken infrastructure: Finally, although getting content for web sites is a common complaint among web application developers [5], this challenge was exacerbated by a lack of infrastructure for getting data out of hurricane-stricken areas. Many shelters lacked computers, and some even lacked phone service for modems. In response, two teams sent volunteers to shelters to relay information and photos using cell phones. Compounding the problem, one HKS creator said that some shelters strangely kicked out these volunteers to protect the survivors' privacy, even though the survivors apparently welcomed the chance to report their whereabouts.

Interpretation: Interviewees were unable to discover and communicate with one another at the inception of their projects. This led to redundant sites. The ongoing absence of a shared task and good communication produced distrust between groups. This inhibited data aggregation as well as data collection from places where in-person data collection was necessary due to infrastructure damage. Section 6 will return to this theme and expand upon it.

4   Motivations of HKS Creators

What motivated interviewees to create and aggregate sites and to endure the difficulties described above? While analyzing the data, we recognized that HKS developers demonstrated a subset of the motivations demonstrated by open source software (OSS) developers, which is reasonable because volunteer developers compose both groups. However, some differences did appear, chiefly the fact that OSS creators demonstrated more self-interested motivations than HKS developers did.

Differences: Certain OSS motivations were not very visible in interviews with HKS creators. For example, whereas some OSS developers created software for their own use and then incidentally shared it as OSS [3], only one interviewee hoped to make personal use of the site (as his sister was lost).

In addition, some developers contributed to OSS projects because their companies paid them to do so [3]. Likewise, in general, most web developers have started projects because it was part of their job [5]. In contrast, only one HKS interviewee initiated his site because he worked for a company with a stated corporate goal of disaster relief.

Several OSS motivations were entirely absent in interviews. For instance, opportunity to learn has been a strong motivator for OSS creation [2] and web development in general [5], but it did not appear to motivate HKS creators, as none mentioned using new skills or tools during HKS projects.

In addition, developers were often motivated by enjoyment from coding OSS, mainly for intellectual stimulation lacking in day-to-day work [3]. In general, some web developers have created web applications for recreation [5]. In contrast, no interviewee mentioned such stimulation or enjoyment—indeed, four handed off some coding to other people, and answering fearful emails was anything but stimulating.

Finally, desires for reciprocal OSS and reputation among peers have motivated OSS developers [2]. HKS creators apparently had little concern for reputation: most did not post their identities in an obvious place, but instead provided anonymous-looking email addresses or "Contact Us" forms. Moreover, none mentioned reciprocity or reputation during interviews. Indeed, one said humility and "not taking credit" was a key to successful collaboration.

Similarities: However, HKS and OSS creators' motivations were not entirely different. For example, many OSS developers valued benefiting "the good of the group" and "helping the cause" [2]. When asked what prompted them to create HKS, all interviewees cited other peoples' needs. Specifically, four HKS creators focused on the absence of "person locator" sites; three made comments like "there was no one system people could refer to." However, although these may be important HKS and OSS motivators, civic-mindedness is not a commonly cited motivation by web developers in general [5].

Moreover, like HKS creators, OSS developers were motivated because success at coding contributed to feelings of achievement or efficacy—"a sense that they have some effect on the environment" [2]. To varying degrees, most interviewees took pride in their achievements. For example, two praised their site's support for heavy query load, and most expressed pride in how quickly their site was completed.

Interpretation: In summary, OSS developers generally demonstrated a larger number of self-interested motivations compared to HKS creators. We suspect that the differences largely resulted from the fact that OSS development has typically involved a long-term commitment of many months or years, whereas HKS development required an intense short-term commitment of only a few weeks aimed at addressing an urgent social need.

Of course, this is not to say that HKS creators were entirely disinterested in their own well-being. However, our interviews do suggest that some programmers may be willing to lead highly draining, sacrificial projects like HKS creation, as long as they know that the engagement will be short-lived and targeted to a significant social need. Section 6 ties this interpretation together with other themes in more detail.

5   Skills, Collaboration, and Success

Because of the wide availability of easy-to-use software such as Microsoft Access and FrontPage, we anticipated that most sites in our sample would be implemented by people with limited technical skill. Yet we knew that some tasks, such as database design, might benefit from the assistance of technically skilled workers, so we suspected that collaboration be a strong success factor.

Interviewees had varying levels of technical skill. Five worked in software development firms or computer science departments; the sixth sold retail goods. Five had created entire web sites on their own in the past; the sixth could edit HTML but had always obtained graphical design help. Three had created databases on their own in the past and continued to do so at least monthly as part of their jobs. Four were managers, one was a student, and one was a graphic designer. All six had 10 to 30 years' experience using computers. One had a BS in computer science, one had an HTML tutor several years ago, and all were greatly self-taught. (All interviewees were male.)

Only two interviewees actually *implemented* the site on their own. (One had database skills; the other did not, so he used a tool that automatically generated a blog and its back-end database.) Two others coded part of the site and then had a professional programmer co-worker finish it, while the remaining two just handed off requirements to programmers who implemented the HKS.

Three interviewees relied on teammates for assistance in *evaluating* what features should be present and whether the site was a viable project at all. In general, interviewees had known their fellow planners through prior shared projects. All six HKS creators relied on existing relationships when choosing where to host their sites; four hosted in-house within their respective firms, and two hosted at firms with whom they had an existing relationship.

These observations indicate that interviewees shored up their skill weaknesses through collaboration but did not develop new trust relationships during HKS projects. Yet this pattern had two exceptions. First, half of the interviewees contacted the media and bloggers by email, phone, or press release to advertise the site; while most contacts with the media relied on prior relationships, some did not. Second, two sites benefited from volunteers who found the site and sought to help. Volunteers answered emails from site users, visited shelters to relay data, typed data into the site

manually, or wrote scripts to populate the database automatically. (Most volunteers apparently lacked the technical skills for this last task.) The aggregator had hundreds of volunteers, so some volunteers set up an email list and Wiki to coordinate work.

A desire for a sense of accomplishment motivated many HKS creators in the first place, so perceived success may affect whether they respond to future disasters. Therefore, after we asked interviewees to state the site's primary goal, we asked whether they felt the site was a success overall; if they said it was a success, we asked why they believed it was a success. To assess if they would "do it all over again," we asked what advice they would give to two hypothetical friends who wanted to create an HKS; one "friend" had no programming experience, while the other had the same programming experience as the interviewee.

In general, the most technically skilled people expressed the most positive views on their site's success. For example, the two most skilled interviewees (who had created sites and databases in the past and coded much of their HKS) both cited high query volume and user testimonials as evidence of success. Moreover, they both would encourage their hypothetical friends to build similar sites.

In contrast, the two least skilled interviewees (neither of whom had created databases in the past nor helped implement their HKS) did not mention visitor testimonials, nor were they enthusiastic about their site's overall success. Moreover, they suggested that their hypothetical friends should collaborate with another HKS creator instead of creating their own. One moderately skilled person was happy with his site's technical capability but said it "just contributed to decentralization" overall.

Interpretation: Several interpretations might explain the pattern described above.

First, prior studies suggested that "developing an application … predisposes an end user developer to be more satisfied with the application than they would be if it were developed by another end user" [3]. Thus, it is possible that technically skilled HKS creators, who tended to be more involved in actually implementing their sites, were predisposed to view their work favorably.

Another reasonable interpretation is that since less technically skilled interviewees relied more on teams, they were more prone to feeling emotionally worn down through friction with other people and through a feeling that the HKS was out of control. This may have colored their view of their site's success. Among interviewees with low self-perceived success, the most commonly reported interpersonal stresses usually related to whether or how to share data with other sites. Stresses also built up within teams: for example, one organization that hosted an HKS did not trust team-mates from other organizations to touch the code on the "live" site, so the hosting organization ultimately became a bottleneck when feature additions were necessary.

A final potential explanation is that existing development tools for building this sort of site required significant up-front learning. After this disaster, there was little time to learn complex new tools. Therefore, HKS creators used whatever tools were already understood or easily understandable (such as blogs) to cobble together a site. Interviewees with more skills could draw on more tools, perhaps yielding a better site. While the data are consistent with this interpretation, it would be desirable to check whether a more objective measure of success, such as number of lives saved or people found, correlated with the HKS creator's skills. However, no such measure is available, to our knowledge.

6 Discussion

Our study highlights several challenges related to developing time-critical web sites in an environment lacking any central coordination mechanism. It appears that these challenges did not result from mean-spiritedness or from leaders' egos: to the contrary, it appears that HKS creators were driven by few of the self-interested motivations that apply to OSS creators. Because concern for "helping the cause" strongly motivated HKS creators, we suspect that most of them would have collaborated with other site creators rather than creating their own site, if they had been aware of each other earlier. Thus, these challenges ultimately owed to the difficulty of discovering other developers' work at an early enough stage to facilitate coordination.

Raising the visibility of projects might also have made it easier for people to match their own skills to specific project tasks—so that people with technical skills could have focused on technical tasks and left counseling to someone more qualified. We suspect that better matching of skills to tasks could lead to more successful projects, further reinforcing workers' motivation to participate.

The key ingredients for HKS creators' problems seem to have been (1) an event that simultaneously inspired several teams to begin projects, (2) the fact that these teams did not know each other when the event occurred and (3) the rapid accumulation of resources (data) that were valuable enough to protect and that were somewhat incompatible between projects. Little about these three ingredients is unique to Hurricane Katrina, so we suspect that similar challenges arise in other circumstances.

For example, we hypothesize that these problems might arise if a large software corporation is caught off guard by a significant event, such as the surprise release of a competitor's product. If the corporation is large and disconnected enough that most of its workers do not know each other, then multiple departments may begin planning responses to the event. These independent projects may yield valuable artifacts such as software code that are difficult to merge later, which might lead to inter-team distrust and infighting rather than collaboration.

In short, we believe that there is a need for improved systems to help workers discover and collaborate with one another early in widely distributed, time-critical projects. Realistically, even with these systems, it is likely that the workers will create some redundant artifacts before discovering one another. However, the systems should help workers discover one another as quickly as possible, in order to keep the number of redundant artifacts to a minimum. Moreover, the systems should help the workers evaluate the quality of whatever redundant artifacts do exist, and it should help the workers merge those artifacts into a unified, high quality whole.

In addition, these systems should facilitate the rapid establishment of trust among workers. Prior research reveals that this requires enabling workers to perceive competent and timely progress by other workers on their shared task [1]. Consequently, the systems should help workers discover and evaluate the work of others. Moreover, to help workers focus on tasks that they will be competent at doing, these systems should help workers to match their skills to tasks remaining to be done.

Existing systems for supporting widely distributed projects do not meet all of these criteria, mainly because they do not facilitate rapid discovery of other workers. For example, there are dozens of web sites dedicated to the coordination of OSS projects;

therefore, discovering whether anybody is working on a certain type of project requires searching all of these sites with a search engine. However, in a time-critical situation, web-wide search engines do not spider these projects quickly enough to be of use. Moreover, teams probably do not bring sites on-line until after investing a substantial amount of work or even finishing their respective sites. By the time a traditional search engine becomes applicable, the teams have already created redundant artifacts.

In closing, our interviews provide a point of comparison for future studies concerning software development after crises. Developers have continued producing new "person locator" sites and similar applications after natural disasters, including hurricanes, typhoons, and earthquakes. It would be interesting to investigate whether leaders of these projects were familiar with the problems that surrounded HKS creation, and if so, whether more recent "person locator" projects have been more successful. If people have led multiple projects of this type, perhaps they have developed promising approaches for dealing with the difficulties outlined in this paper. Studying their experiences may provide additional perspectives on highly distributed, time-critical software development and generate more data to support deeper analyses yielding new solutions to these challenges.

7 Acknowledgements

References

1. Jarvenpaa, S., and Leidner, D. Communication and trust in Global Virtual Teams. Organization Science, 10, 6 (Nov-Dec 1999), 791-815.
2. Lakhani, K., and von Hippel, E. How Open Source Software Works: Free User to User Assistance. Research Policy 32, 6 (June 2003), 923-943.
3. Lakhani, K., and Wolf, B. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In Perspectives on Free and Open Source Software, MIT Press, Cambridge, MA, 2005, 3-22.
4. McGill, T. The Effect of End User Development on End User Success, JOEUC 16, 1 (Jan-Mar 2004), 41-58.
5. Rosson, M.B, et al. 2005. "Designing for the Web" Revisited: A Survey of Informal and Experienced Web Developers. In International Conference on Web Engineering 2005 - Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2005, 522-532.