Activity Traces and Signals in Software Developer Recruitment and Hiring

Jennifer Marlow
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA
jmarlow@cs.cmu.edu

Laura Dabbish
Human-Computer Interaction Institute &
Center for the Future of Work, Heinz College
Carnegie Mellon University
dabbish@cmu.edu

ABSTRACT

Social networking tools now allow professionals to post and share their work in online spaces. These professionals build reputation within a community of practice, often with the goal of finding a job. But how are the visible traces of their actions and interactions in online workspaces used in the hiring process? We conducted interviews with members of the GitHub "social coding" community to understand how profiles on the site are used to assess people during recruitment and hiring for software development positions. Both employers and job seekers pointed to specific cues provided on profiles that led them to make inferences (or form impressions) about a candidate's technical skills, motivations, and values. These cues were seen as more reliable indicators of technical abilities and motivation than information provided on a resume, because of the transparency of work actions on GitHub and relative difficulty of manipulating behavior traces. The use of online workspaces like GitHub has implications for the type of information sought by employers as well as the activity traces job hunters might seek to leave.

CATEGORIES AND SUBJECT DESCRIPTORS

H.5.3 Group and Organization Interfaces – *Computer Supported Cooperative Work*

GENERAL TERMS

Design, Human Factors

KEYWORDS

Impression formation; Transparency; Hiring; Open source software development; Impression management

INTRODUCTION

Reputation and professional development have long been cited as primary motivators for contributing to online peer production communities like open-source software development projects [16]. Contributors and potential employers both view open-source software participation as a desirable way to gain relevant experience and skills [17, 23]. The pace of recruiting and career development in online production communities seems to be accelerating with the emergence of social media, evidenced by discussions among developers online sparked by blog posts such as one asserting that "GitHub is your new resume" [3].

In many online peer production environments, social networking functionality is now tied directly with the work artifacts being shared or collaboratively developed. This means contributors can get moment to moment updates about others' actions on artifacts and interactions, affording an unprecedented level of transparency around who is doing what and how work is accomplished. At the same time, there is some level of individual control over what is shared and how.

Anecdotally, the open source software community describes the traces provided by social media as affording more verifiable information about an individual's skills and abilities than a list of achievements on a resume. As John Resig, the creator of the jQuery interface library, recently tweeted: "When it comes to hiring, I'll take a Github commit log over a resume any day."

Social networking functionality, when tied with the work environment in a peer production site, provides moment-by-moment information about actions on artifacts and interactions around project decisions or activities. This means that information in these environments, compared to information on a resume, can provide much more information about how someone works. In software development, for example, sites like GitHub allow employers to view the details of the code an individual writes in each commit, or contribution, to a project, and any interactions or discussions around the code are also publicly viewable. Potential employers can effectively reconstruct exactly what someone works on, how they work, what their code looks like, how they talk about their work or negotiate changes to collaborative projects, and their speed and style of work on public projects. This level of detailed information about someone's working style is typically unavailable to a potential employer.

Research has not addressed how activity traces within online communities of practice play into the hiring process for more traditional jobs associated with the work of the

CSCW '13, February 23–27, 2013, San Antonio, Texas, USA.

community. Thus, we were interested in how employers in this environment made use of the displayed cues about developers' actions over time. Our work focused on the following research questions:

(1) How are activity traces in an online peer production community used by potential employers to find and evaluate prospective software development hires?

(2) How do job seekers attempt to manage the impressions their activity traces give off to employers?

We conducted exploratory interviews with employers and job seekers in GitHub, an online open source software hosting repository with extensive social networking functionality integrated with the development environment. This means that potential employers can view an individual's profile of projects posted on the site, and a history of their code related actions on these projects and other people's projects over time. As software developers commit changes to their software projects, these changes are broadcast to other developers watching the project. A history of commits (or contributions) to the code is recorded over time, along with conversation around changes in the form of comments. (See [5] for a detailed description of the GitHub environment.)

We interpret our interview results through the lens of signaling theory to understand how and why certain cues were viewed as reliable signals of underlying characteristics of a potential hire. Our results suggest that certain activity traces are viewed by employers as more reliable than others, in part because they serve as more trustworthy signals of underlying characteristics that are often difficult to assess in traditional interviews, such as values and motivation, but also because they are easy to verify quickly. For example, willingness to actively contribute to software and share projects openly on the site was viewed as a reliable signal of commitment to open source software ideology.

BACKGROUND

Impression formation and employment

When evaluating job candidates, either online or offline, a principal goal of the employer is to accurately evaluate applicants' job-relevant knowledge, skills, abilities, and other characteristics. These evaluations, in turn, affect selection decisions [2]. For example, related work in the online peer production realm looking at admin permission granting in Wikipedia revealed that reviewers weighed evidence of interaction style, a candidate's social network, and the amount and type of past editing work when making these decisions [6].

In addition to criteria relating to competence and expertise, employers also often place high importance on features that cannot be gleaned from a resume and need to be assessed in a job interview, such as a person's likeability and the potential person-organization fit [28]. Other work suggests that features such as whether a candidate is seen as "the right person for a job" and perceived applicant-interviewer similarity can be strong determinants of hiring decisions [14].

Increasingly, online presence on social networking sites (SNS) such as Facebook or LinkedIn are playing into the hiring process for full time jobs in the offline world. A 2008 survey [13] found that information gleaned from SNS could damage an applicant's chances of being hired if it revealed that a person had lied about their qualifications. On the other hand, SNS information could be advantageous if it helped support their qualifications or portrayed a professional image. SNS members with personal profiles are aware that employers might look at their profiles [2] and occasionally engage in management techniques to present a professional image [7].

While it is known that employers supplement resumes with online information and use this to form impressions about candidates, less is known about what specific inferences hiring managers make from this information and how useful or accurate these impressions actually are. In the software development domain, recent surveys indicate that employers examining students' OSS experience look for compatible skills (while paying less attention to the popularity of the projects,) [17] but it is unclear how exactly they go about this. This work has not considered how activity trace information would play into this process. In the next two sections we consider the impression formation and impression management process from the perspective of signaling theory.

Impression formation as signal assessment

Signaling theory provides a useful framework for understanding impression formation in the hiring process. According to this theory, we make assessments of others based on their visible characteristics and actions. These observable cues effectively act as "signals" of hidden qualities (such as experience or expertise) that are not directly observable [9]. This theory delineates two main types of signals in terms of how they are produced and interpreted: *assessment signals* are thought to be more reliable indicators of the presence of a certain quality because they are costly to produce, whereas *conventional signals* are more susceptible to being manipulated because they are more easily faked by someone not possessing the underlying quality they signal. For example, being able to lift a heavy weight is an assessment signal of someone's strength while simply wearing a Gold's Gym t-shirt would be a conventional signal of this underlying quality (something that can easily be acquired and worn even if the wearer is actually quite weak) [15].

In the online realm, signaling theory has been applied to examine how individuals in online communities attempt to convey and interpret visible cues about others as signals of their underlying characteristics (e.g. user name as a low cost or conventional signal of interest in the topic of the community) [8]. This work suggests that the degree to which a certain type of cue is viewed as reliable may depend on the context of the site: For example, the interpretation of one's number of friends or connections on a site as a signal of popularity may be more or less reliable depending on how costly it is to make a connection [9]. An additional important aspect of evaluating others' signals is the amount of effort involved, or how easy it is for the observer to verify the accuracy of these signals [18].

The impression management and formation process in online peer production communities, then, can largely be thought of as a signal production and evaluation process. Job seekers on the sites can attempt to convey or signal certain skills or abilities with the information they post on their profile or the activities they engage in. Employers must determine which visible signals of developer expertise or personality to attend to on these sites (depending partly on how hard it is to verify them), and then interpret these signals to infer the developers' underlying or actual skill or expertise.

In the hiring domain, where deception about qualifications is a concern, being able to judge signal reliability is important. When the costs of forming an incorrect impression are high, for example, in hiring for a highly-paid job, perceivers may demand a more reliable signal that is costly to fake [9]. Level of education attained is one example of a reliable signal of skills that is costly for a person to produce [22] but also potentially costly for an employer to verify. Closely related to the issue of signal evaluation is the issue of signal production, or impression management.

Impression management

Signaling theory also has important implications for impression management, because signalers may deliberately try to convey positive attributes to receivers. Given that signalers may have incentives to "cheat" [4], understanding when and how they do this (and how receivers go about verifying the signals they produce) is an important topic that we investigate in GitHub.

Initial work by Goffman [12] focused on ways in which individuals convey information about themselves to observers, which can be the "cues" they intentionally give, or their real behaviors, which may be "cues given off" through involuntary expressive behavior. Both types of information can be manipulated, either through overt deceit or through pretending. In the online realm, studies of impression formation and self-presentation in online settings have examined the cue management process in a variety of contexts and scenarios, ranging from honesty and lying about oneself in online dating profiles [10, 11] to friendship formation and other behavior on sites like Facebook [7, 15, 27] to blogging [26]. This work has primarily focused on understanding self-presentation in the context of interpersonal or non-work relationships, although Ellison et al [10] draw parallels between online dating profiles and resume submissions for jobs, as using deception in either arena can be grounds for terminating a relationship. However, work on impression management in the social realm has focused largely on how people attempt to control the impressions conveyed by their profiles or pictures on these sites, and is not centered around impressions relating to work artifacts, skills, or behaviors.

Given that site design affects the reliability of signals [9], what signals do employers attend to in a peer production environment providing a plethora of trace information about work process and collaborative activity? How do prospective employees manage these signals? In order to understand how this new set of information plays into hiring, we examined how activity traces were used as signals in an online peer production environment instrumented with social media. We wanted to understand how these traces influenced employer impressions of potential candidates and how candidates attempted to manage these impressions.

INTERVIEWS OF GITHUB USERS

Method

We conducted a series of semi-structured interviews with thirteen GitHub users to identify how activity traces are used and assessed to infer a developer's abilities and personal qualities. We began by sending a screening and recruitment questionnaire to 200 GitHub members with publicly available e-mail addresses on their profiles. As there was no way to specifically filter for our two target groups (university students and employers,) we focused on targeting people located in North American and European cities that were likely to have large populations of both technical students and companies (e.g. San Francisco Bay Area, Boston/Cambridge, Pittsburgh, Seattle, Waterloo, Toronto, London, Berlin).

The questionnaire asked people if they had ever used GitHub as part of the job application or hiring process and if they would be willing to participate in a follow-up interview on the topic. Overall, 128 people responded, 65 of whom volunteered for the follow-up interview. Participants did not receive any compensation for taking part in the study.

We contacted respondents for interviews in the order in which they replied, sampling both employees and job seekers in order to understand the hiring process from both sides. Our participants for these interviews were seven employers who reported using GitHub to identify and evaluate job candidates and six job seekers who reported using their GitHub profile to supplement their job applications. The interviews focused on how they had used GitHub during a recent hire or job application.

We asked employers to describe a recent past hire, focusing on how they used GitHub during that hiring process, what information on the site was attended to and what that information conveyed about the candidate. We asked job seekers to describe how they used GitHub, how it had played into any recent job applications or interviews, and whether and how they edited the information on their profiles or in other public places on the site.

In our analysis, we coded the interview transcripts to identify the different ways profiles were used in the hiring process, as well as the different types of inferences made about individuals being evaluated based on 'signals' in the GitHub environment. Using HyperResearch, a qualitative analysis software tool, we identified relevant sentences or broader segments in interview transcripts related to candidate evaluation, and then open-coded these segments for comments related to profile cues and inferences made from them. Next, specific instances of these themes were compared across interviewees and further refined as necessary, until a set of recurring themes about signals and the inferences drawn from them emerged. The interpretation of these from both employer and job-seeker perspectives is addressed in detail in the following sections.

EMPLOYER PERSPECTIVE

The employers (6 males, 1 female, referred to here as E1 through E7) all worked for software-related organizations based in the United States (both large and well-known internet companies and smaller startups.) These companies varied in size: three of them had less than 50 employees, two had between 200 and 500 employees, and one had more than 500 employees. On the survey, participants indicated that they had either asked job seekers to provide links to their GitHub profiles during the hiring process, or had actively searched for people on the site to recruit and/or learn more about them. The interviewees were asked to think aloud while consulting profiles of people they had hired or were thinking of hiring. We used this method to obtain detail on how information in the GitHub environment signaled developer characteristics.

Employers' use of GitHub

Our first research question focused on how employers use GitHub profiles to evaluate new hires. All interviewees expressed the belief that a GitHub account provided insight into an individual's technical abilities and/or personal qualities in a more reliable way than resumes or code samples taken out of context. The GitHub profiles provided employers with a history of the individuals' contributions over time, and further guarantee the candidate was indeed the author of any code submissions.

Table 1 summarizes the main GitHub signals and inferences mentioned by employers in the interviews. We categorize these cues based on factors that are relevant to both the profile holder and profile viewer: Signal type/reliability, and the ease with which the viewer can verify them.

In the rest of this section we describe in detail how employers used GitHub activity traces as signals of a job candidate's motivation, quality of code contributions, and soft skills or management abilities.

Signal	Inference	Signal reliability, ease of verifiability
1. Active open source involvement	Shared open source values	Reliable Easy
2. Contributions accepted to high status project	Community acceptance of work, quality of contributions	Reliable Hard
3. Project ownership	Soft skills: Initiative, project management	Reliable Easy
4. Side projects	Passion for coding	Reliable Hard
5. Number of watchers or forks of project	Project popularity	Unreliable Easy

Table 1. Summary of employer inferences from profile signals

Inferring motivation

Employers in our sample worked to assess how well job candidates would fit with their company or team culture (or person-organization fit). These factors are traditionally assessed during interpersonal interaction in face-to-face interviews through direct questioning. The employers in our sample indicated that a job seeker's profile of activity on GitHub signaled personal characteristics of the employee such as being a team player, showing commitment to their work, or demonstrating how he/she spent their free time.

Shared open source values and character

Employers care about value congruence with their employees. In the software development world, an important and hotly contested value is attitudes towards open source and whether software should be free. In fact, there is a well-documented ideology of open source software [24]. Developers who differ in their software ideology may thus be said to come from different cultures. Thus, an important character property of a developer is their attitudes and commitment to the open source ideology.

Employers in our sample used presence on GitHub and activity levels on the site as signals of the level of commitment to the open source ideology. Simple presence on GitHub (having a profile and sharing even one repository) was viewed as an indicator of a potential employee's open source values by four of seven employers. The presence of code that was developed openly and shared with others signaled even more strongly that the developer valued openness, transparency, and participation in a community. This active involvement in the open source community was a signal of the candidate's selflessness and honesty. As one employer put it:

> *"If they've devoted time to this OS project, that's a good indicator that they're in [computer science] for*

> *the right reasons. Software engineering is becoming a pretty lucrative career…you could liken that to a doctor working with Doctors Without Borders. They're doing something because they want to give back to their community*" (E7).

Active participation in other people's projects was the most reliable signal of commitment to the open source mindset (mentioned by five out of seven employers). Cues such as recent and frequent commits in another person's project showed the candidate was indeed invested in the open source community (E2, E3, E4, E6, E7). It is rather trivial to create a profile on GitHub and fork other users' projects (meaning create a personal copy of the project in order to make changes to it). Having a copy of someone else's project did not signal investment. The effort to fork a repository was negligible, while the effort and skill required to contribute meaningfully was much higher. This was widely understood, as one respondent described:

> *"a lot of people will just fork a lot of projects kind of to collect them but not actually do anything with them. So I look for a sign that these are things he's genuinely engaged in*" (E4).

Activities within these forked projects were costlier signals of commitment, requiring much more effort to produce. This activity, publicly building on another person's work, served as an assessment signal that the candidate truly bought into the open source mindset (over and above mere presence on GitHub or simply forking projects). As one employer explained:

> *"[by looking for recent activity I was] sussing out whether they're a good sport about contributing to open source…if they're doing their job of keeping up to date and actually participating" (E2).*

Passion for programming

Organizations also differ in their working style or company culture around work life balance. Employers in our sample wanted to assess candidates' level of dedication to the work and their level of initiative. They were able to discern subtle motivational differences that suggested person-organization fit from the kinds of projects a developer worked on.

Our interviewees described going through an individual's public repositories to figure out how they spent their time outside of work. They categorized projects on a user's profile as either work-related repositories that were part of an individual's "day job" (work or schoolwork,) and non work-related side projects, which could either be contributions to open-source projects not directly related to work, or personal projects done as a hobby or for fun but not necessarily intended for a wider audience.

For many of our employers, personal projects signaled a candidate's love for programming and willingness to do it in one's leisure time as well. One interviewee (E4) saw personal projects as a signal of interest in learning and developing one's career, while two employers (E3, E5) described using this signal to assess whether the candidate shared the same enthusiasm for coding with the other members of their organization and were the type of people they liked to work with. This signal is valued because it ties into the aforementioned tendency for employers to like and seek out people who are similar to themselves and fit with their company's culture. As one employer explained:

> "*A lot of us spend our weekends working on [project name] so we want to work with people who are motivated to not just work on the code they've been assigned but to work on projects outside their job. It just shows a general excitement for the space and that's what we want to find – people that are really engaged"* (E5).

These side projects suggested a willingness to learn and revealed excitement about the software development domain. For employers this meant a potential employee who would spend their free time working, and show initiative and entrepreneurship in their work.

Inferring quality of contributions

Employers also care about a potential hire's competence and level of skill for the job. GitHub supported traditional methods of evaluating software development, allowing employers to look directly at the content of someone's code and the languages they had used. The cross-project visibility and the community on GitHub supported skill assessment beyond these traditional uses. Specifically, affiliation and accepted contributions to a popular project reliably signaled candidates' level of coding ability in the GitHub environment.

Accepted code as a seal of approval

If a candidate had contributions accepted to well-known open-source projects, it was seen as a community-level seal of approval. An accepted commit to a high-status project (a widely-used project with many contributors and watchers) signaled the candidate was someone who produced quality code. This acted as a reliable signal because it required approval of the code by others in the community, meaning it would be extremely difficult to falsify. For example, one employer described a candidate who seemed proficient because he had committed code to a high-status project:

> "*Seeing that he had commits to jQuery, was filing tickets with jQuery, and I know that's a prestigious project to work on…Just by looking at his code, if nothing else seeing that it was being merged downstream into jQuery, I recognized that has demonstrated some level of proficiency*" (E6).

Another interviewee echoed this view for candidates who had contributed to open source projects, likening it to a reference:

> "*someone else can vouch for your work because you were good enough to work on that project, be a part of that community*" (E7).

Since examining lines of a developer's code can be a time-consuming endeavor, using the reputation of previously-established projects that had accepted an individual's contribution as a proxy for quality (or lack thereof) was one way to reduce the evaluation costs of the perceiver in forming impressions about the abilities of a coder. An employer's opinion of a project's reputation in our sample was largely based on general knowledge of the wider community or past experience with its use versus visible cues at the project level such as watchers (people who have decided to "follow" the activity of a project) or forks (people who have saved a copy of the project to edit on their own.).

Popularity does not always equal quality

Employers in our sample also noted conventional signals of quality they did not trust. Primary among these were popularity signals: simple counts of watchers on a project or followers (people subscribed to a developer's activity feed). In some community settings, indicators of popularity (such as the number of votes given to an answer on a question-answering site) can serve as a proxy for the quality of the answer, while in other settings, popularity (e.g. having too many friends on a SNS) can be viewed negatively [25].

Project popularity on GitHub can be roughly assessed by the number of other people "watching" the repository along with the number of people who had forked that repository. Only one employer (E2) specifically mentioned looking at a candidate's main project to look for a large number of forks. Two employers (E4, E7) were more skeptical about the utility of the watching/forking numbers as indicators of a developer's ability. As one interviewee explained:

> "*I don't think I've hired or recruited someone specifically because they were working on a very popular project or something. There aren't enough popular projects and the popularity doesn't necessarily indicate quality for that to work."* (E4).

Popularity was thus to some degree viewed as a signal that developers could game. Our interviewees noted that project popularity was an unreliable signal of code or developer quality because it had more to do with how much an individual promoted their work:

> *"You can see if a lot of people have watched and forked and that's a good thing, but it kind of depends on how good a marketer that person was as well on GitHub."* (E7).

Rather than relying on numbers of forks and watchers, employers described looking at the project where the applicant had made the most commits (as presumably that was the work they were the most serious about or interested in) and then assessing the actual code that was written there to understand the individual's style and skill level.

Inferring developer "soft skills"

Finally, one employer mentioned inferring "soft skills," such as project management abilities, through a developer's activity traces. These cues were largely gleaned from their interactions in a project they had started rather than one they had forked (a distinction which is made obvious by the GitHub interface).

These projects allowed him to observe a candidate's project management skills and collaboration style. Owning a project involves tasks such as setting a design direction, managing incoming code contributions and patches, and interacting with potential collaborators. The employer described being able to infer these soft skills from projects that a person owned, stating that

> "*[Projects he owned] would give me a better sense of his long-term design skills and ability to manage a community"* (E4).

JOB SEEKER PERSPECTIVE

Our second research question was how job seekers manage their presence on GitHub. In order to address this question, we also conducted interviews with six job seekers on GitHub (referred to here as J1-J6) from four different U.S.-based universities. These participants reported using GitHub to supplement their application materials as a job seeker in our recruitment survey. Of these, two were undergraduate students in their final year, one was a PhD student, and the other three recently graduated and had been working at their current position for one year or less. They had between 4 and 28 public repositories visible on their profiles (mean= 17.3) and an average of 19 followers.

All of the interviewees had originally joined the site to host projects they were currently working on and archive past work. The projects they shared included school and class projects, projects for hackathons, samples of code written for job interviews, side projects, and open source projects, both recent and older.

They varied in the ways they shared their work on GitHub with employers. Four of them had provided a link to their GitHub accounts on their resume and stated that this had been consulted during the hiring process. One person never provided this information to the employer and instead assumed the employer had found his profile through a Google search. The final interviewee did not actively promote his GitHub account in the hiring process but stated that he would provide a link if asked for it.

In the interviews, they described the role that their GitHub account had played in recent employment searches. We asked them to describe the impressions they thought their profiles conveyed to potential employers. Finally, we talked about how they managed their profile for an actual (or imagined) audience of potential employers.

Impressions given from profile

We asked interviewees what impressions they thought their GitHub profiles conveyed about them to a potential employer. Their responses highlighted two key qualities they felt employers could see from their activity on the sight: passion for software development as an activity and field and technical expertise.

Activity signaling passion for the field

Interviewees mentioned that their visible level of activity in the site was a positive signal of passion for software development (J1, J2, J6). One person realized that it might be important from the employer's perspective to see evidence of extracurricular coding activity. Referring to some of the side projects, he mentioned,

> "*The fact that in my free time I went and coded something…I think that counts a lot"* (J2).

Our interviewees also indicated their activity in terms of following also signaled an interest in the domain. The fact that they followed a lot of projects conveyed that they were generally interested in keeping up-to-date on the latest projects and technology (J2, J3). One interviewee followed several hundred coders and projects. He thought this would show employers his interest in coding and open source even though he himself had very few projects. Another explained,

> "*The impression I would like to give off is: This guy is current with what's going on in the open source world"* (J2).

Portfolio of work signaling technical expertise

Developers were also keenly aware that the portfolio of projects on their profiles conveyed their areas of expertise to employers. One person's profile (J5) contained a lot of projects in "web-heavy" languages like Python and JavaScript. He thought revealed that his skill set focused on web development. Several interviewees noted the languages they used in their projects would signal proficiency in those languages (J4, J5, J6).

Interviewees were also aware that details of their code would influence perceived skill level. They noted that good style such as descriptive commit messages, comments, and code indentation would convey competence. Two interviewees were aware that their code was not particularly readable or well-organized, partly because the projects were originally just intended for personal use and trying new things. One person had many old projects on his site and his style had improved since. He explained,

> *"I have a lot of things in my GitHub account that I wouldn't feel comfortable putting on a resume because I didn't know how to code at the time…I'm definitely a better coder than some of the projects imply"* (J5).

Impression management

Attitudes towards profile manipulation

Although the job seekers in our sample were acutely aware their profiles gave off certain impressions about them, few reported having engaged in efforts (such as changing or editing any information visible on the profile) to control this impression. We learned in our interviews that this was partly because there were few ways of doing so and doing so was too costly and effortful. As one interviewee explained:

> "*A major component of your GitHub profile are your projects, and it's not really easy to refactor your code quickly"* (J1).

Job seekers also noted that even old or incomplete material could still be useful in conveying qualities about its creator. One interviewee intentionally left incomplete or unfinished projects on the site because they could signal thought processes such as "*how you are approaching and organizing something"* (J4).

Signal Accuracy: Cleaning up the profile

The developers we interviewed realized there might be some need to edit information on their profile, either currently or in the future, to more accurately represent their work. This profile improvement consisted of either commenting code or removing old projects. For example, one developer expressed a desire to concentrate more on "cleaning up" his profile. In lieu of having the time to improve the accuracy of the signals on the site, he provided employers with a description of the current state of projects in the repository descriptions:

> "*I try to say statuses of projects sometimes: I have a couple repositories where it says something like 'this is a super early proof of concept thing'…I make a slight effort to sort of downplay the things I'm not very proud of"* (J5).

He mentioned that in the future, he would consider stripping out everything that he was not proud of, including old and incomplete projects, in order to provide a more accurate picture of his current abilities.

Overall, we found that participants did not seem to engage in much "gaming" of the system for a variety of motives. On one hand, they often felt that they were happy with the signals given off by their profiles regarding their active involvement and participation in projects. On the other hand, they did not try to fake information because it was effortful (i.e. difficult to give a false impression about actual code they wrote.) They could influence information given off by projects by downplaying them in the repository descriptions or by adding proper comments into the code. Keeping the profile as-is, with imperfections or old information, was seen as useful, both to other developers (who might be able to learn something by viewing old projects) and to employers.

DISCUSSION

Our interviews revealed that employers used cues on GitHub as signals of underlying characteristics of potential hires. These signals varied in their reliability as a function of manipulation cost in the GitHub environment. In addition, employers' use of these signals was directly related to evaluation cost. Job seekers on the other hand, were aware that employers were using their behavior as signals of underlying properties. In addition, attempts to manipulate these signals were culturally or practically discouraged. We consider what made signals reliable and how evaluation cost influenced employer behavior.

Evaluation of signals and inferences

Reliable signals

Employers expressed a belief that some cues were not easily manipulated and thus they were reliable indicators of a potential hire's competence, values and collaboration styles. Our analysis revealed that two key properties of activity traces supported signal reliability. These were the fact that traces were: (1) archived over time - signals generated by direction of time and effort expended (commitment to open source values, passion for programming), or (2) networked - signals generated by third party affiliation.

Activity traces on GitHub are archived and persistent, meaning employers could view a history of effort on projects over time. Employers could use information about how much effort was put into different projects to draw conclusions about a person's values and motives. This signal also appeared to be an accurate indicator of motivation. For example, job seekers who committed to open source projects truly wanted to give back to the community. This kind of historical signal would be extremely costly to fake, making it a more reliable indicator of investment.

The network context of activity traces in GitHub meant that affiliation could also act as a signal. The acceptance of offerings to joint projects in the form of pull requests or code commits was archived and accessible. The fact that work had been accepted to well-known projects was not easy to fake. Connection to these high status projects increased employer perceptions of the candidate's competence or quality. These signals were deemed more trustworthy because they were generated by a third party. This is akin to the notion of warranting introduced by Walther et al. [27]. This work suggests that when forming an impression, information provided by third parties (for example, the fact that someone else had accepted the candidate's code into their work) and connection to a social network is more trustworthy than self reports. The offering and acceptance of work contributions provided a kind of third party verification of work quality.

Unreliable signals

Activity traces were deemed unreliable signals of a potential hire's characteristics if they were (1) subject to manipulation or (2) likely to be missing information because of the way they were generated. Both of these assessments required employers to have intimate familiarity with the GitHub environment and the social context surrounding project activity.

Project popularity was an example of an unreliable signal of a potential hire's ability. Although popularity was easy to assess through aggregate statistics on the number of people watching a project, it was seen as subject to manipulation and thus not closely tied to ability (the underlying characteristic employers wanted to assess). Project popularity could be artificially inflated by an applicant, and was dependent on factors distinct from coding abilities such as how well an individual marketed his or her project or what type of project it was.

Signals were also deemed unreliable if they were known to be missing information. For example, employers acknowledged that projects on an individual's profile were not always representative of their full body of work. This meant they were only partial signals of properties like technical skills and coding abilities, since private projects were not visible on GitHub. If the majority of an individual's coding activity happened behind a firewall at work (meaning they could not share it), then their GitHub account would not completely reflect their true range of abilities.

Employers were aware that certain information about a potential hire was not visible on the profile and the possible reasons it might not be visible. Despite this knowledge of the broader context or perhaps because of it, employers in our sample valued an active presence on GitHub. There was a sense from the employers we sampled that all things being equal, a candidate with a GitHub profile would still be preferred because of the added signals. One person said,

> "*I don't think you can use it as the sole way to judge someone because a lot of it's going to depend on how much time the developer is donating to maintaining their GitHub, so if this is someone who has a wife and kids and not much time to be programming out of the workplace, that doesn't necessarily mean they are not as good of a developer. But if something is up there, it's definitely a huge plus and probably one of the first things we look for – are they sharing their source code?*" (E7).

Evaluation cost

Evaluation cost seemed to influence employer's use of activity traces in GitHub. The signals summarized in Table 1 are all observable on a user's profile on the site. It is easy to quickly verify active open source involvement just by seeing that a person is a member of GitHub and has evidence of recent activity in their profile's activity feed. It is also easy to tell whether a project was original or forked, via the presence or absence of an icon.

Determining contributions to a high-status project required more effortful investigation if these commits were not in the top recent activities shown, while determining if something is a "side project," as one person mentioned, may require following up with a user to learn more about it (depending on how the repository is described in its textual summary.)

While one possible advantage of GitHub signals is the ability to view how people work with others, interviewees did not mention looking at evidence of past interactions with others to determine attitude and personality. Work on hiring has shown employers are usually interested in candidates' likeability as a team member. In other contexts such as Wikipedia, traces of civil online interactions are used as a signal of likeability and collegiality [6]. However, in GitHub, it is non-trivial to view an individual's interactions. This information requires more effort to seek out and is time consuming to review.

Site design strongly influences the cost to access information about a person. This cost may have repercussions on what information feeds into the hiring decision. The employers we interviewed favored cues that took less effort to access and verify, such as relative effort across projects or presence of forked versus owned repositories. They used these cues as a heuristic to identify people with extra passion in addition to skills, who were then set apart from general applicants or followed up with in person.

Impression management

Our second research question focused on impression management by job seekers. Participants acknowledged that they could not easily give a false impression about the actual code they wrote, but they could try to influence information given off by their projects through downplaying them in the repository descriptions or adding more proper comments into the code. Given that GitHub has relatively recently gained popularity as an employment tool, it is possible that future site members will be more conscious about managing their impressions for future employers.

Design implications

Our results have implications for the design of open transparent work environments. The effort required to access activity traces connected to a person may change what is considered during the hiring process. Our results suggest it is important to strike a balance between providing enough information to be useful while avoiding overload. This could be useful in other online peer production settings such as Wikipedia: Visualizations of an editor's work history across pages could help editors understand what kinds of tasks they would be good at performing. This kind of history could also give context to debates and arguments.

Our work also raises interesting questions about designing activity traces for multiple audiences. GitHub is a site that was not originally designed as a hiring tool. Its design also needs to support collaborative software development. Activity traces are used by developers to coordinate their work and transfer knowledge across projects. However, employers increasingly use these traces to evaluate new hires. There is a tension between using the site as a tool to carry out work and using it to manage the impression one gives to external observers.

The degree of overlap between signals that are important to manage for gaining acceptance within a given community and signals that are important to manage for external viewers such as employers or evaluators is another area that future work can tease apart and provide insight for transparent system design. As Begel et al [1] suggest, a social medium that reveals a user's knowledge, expertise, activities, or availability may be useful for finding knowledgeable others when consumed by the user's peers, but may feel like "corporate spyware" when consumed by his manager. Future systems could provide different views or privacy settings for different categories of users, while still providing the benefit of transparency of actions and reliable assessment cues.

Implications for other domains

The importance of passion for coding and active involvement in the community may be uniquely emphasized in GitHub due to the nature of the open source development community, in which people may be more motivated by self-development and reputation gaining compared to contributors to online content sites such as Wikipedia. In addition, contributing software to open-source projects also requires a certain level of expertise to pass the review process, compared to Wikipedia [19].

However, some of the issues raised here may also apply to issues pertaining to the role of publicly visible behavior in other realms outside of GitHub. For example, there are other instances in which people in technical fields may wish to promote their skill or knowledge in publicly viewable online arenas.

While there has been much concern about employers accessing individuals' personal SNS profiles, there are other areas in which employers may wish to access public work-related information, and sometimes job-seekers may wish for employers to see this information. For example, there has been discussion in the developer community about when/if one should display their StackOverflow score on their resume, including varied reactions from employers about the meaning of this score (e.g. [21]). Users of the Quora question-answering site have also mentioned that they will strategically answer questions in order to attract the attention of influential people and as a form of networking [20].

As people interact and collaborate online on sites geared towards "serious leisure," they may generate additional traces as they go about their work. It is possible to imagine a future of work in which evidence of one's abilities and what "type of person" they are is not solely gathered from their activity within a structured organization or educational program but from the aggregation of their behavior on the web. For example, are the details of a person's online Wikipedia editing activity considered to be useful by employers looking to hire journalists or writers, and do Wikipedia contributors looking for jobs actively promote the work they have done on the site? These questions suggest interesting research directions to explore.

Limitations and future work

Our sample of interviewees was limited to a small subset of people who volunteered to discuss the topic of GitHub and hiring. Another question that was not deeply addressed in the current research was ways in which impression formation occurs with other cues and information besides a GitHub profile. It would also be valuable to follow up on employers' perceptions of new hires after they have spent some time in their job role to understand how initial impressions correlate with actual performance, or how these change over time.

Future work can also include developers and employers who are not involved in the GitHub community or who participate in other sites such as BitBucket or Sourceforge to understand if they present or market themselves differently in these environments, as well as to understand why they choose not to use GitHub.

Furthermore, the role of impression formation and management in other sorts of online production communities should be examined, to uncover similarities and differences across domains outside of open source software development.

CONCLUSION

In this work we examined how activity traces on GitHub are used as signals by potential employers. Past work on impression management online has suggested people actively manipulate various observable elements of their online presence to portray an idealized self. Interestingly, we saw very little evidence of such behavior in GitHub. This may be because the effort required to manipulate activity traces that actually matter to employers (such as active participation and code style) is too great.

At the same time, employers preferred easily verifiable signals of potential hire's skills or abilities. This desire to minimize assessment effort meant they did not necessarily take advantage of the full range of transparency that GitHub can provide. They used candidates' presence and high level cues of activity on the site to infer how well the person would fit within their organization. Employers favored developers who signaled similar passion and willingness to work on coding projects both during and outside of work.

The kind of transparency pioneered by GitHub may have implications for the future of hiring. Employers may begin to expect applicants to provide a rich history of detailed work activity traces. Job seekers may in turn increasingly gravitate towards companies that will allow them to accrue a publicly-available (or shareable) portfolio of work. Businesses may find employees demand some open-source work sharing, and see policies of openness as a benefit. We are already seeing this kind of trend in fields like graphic design, as individuals often work for themselves to maintain the ability to visibly promote their work.

The implications of our results extend beyond software development as work becomes increasingly digital. Providing accessible, reliable traces of an individual's work history may support more accurate impressions of unknown contributors. These impressions will shape decisions about recruiting, hiring and promotion in tradition and new forms of organizations like Wikipedia or crowdsourcing. Such impressions are also likely to influence the collaborative dynamics of work. It is important for system designers and policy makers to consider what actions and activities can and should be recorded and made visible. Our results should help decision makers develop useful and efficient ways of providing various groups with the information they need while protecting individuals' rights.

ACKNOWLEDGMENTS

This work was supported by NSF grants IIS-1111750, SMA-1064209, OCI-0943168, grants to Dabbish from the Center for the Future of Work, Heinz College, Carnegie Mellon University, and an NSF Graduate Research Fellowship.

REFERENCES

1.Begel, A., DeLine, R., & Zimmermann, T. Social media for software engineering. *Proceedings of the FSE/SDP workshop on Future of software engineering research*, (2010), 33–38.

2.Bohnert, D., & Ross, W. H. The influence of social networking web sites on the evaluation of job candidates. *Cyberpsychology, Behavior, and Social Networking*, *13*(3), (2010), 341–347.

3.http://code.dblock.org/github-is-your-new-resume (Retrieved Aug 25, 2012).

4.Connelly, B.L., Certo, S.T., Ireland, R.D., and Reutzel, C.R. Signaling theory: A review and assessment. *Journal of Management 37*, 1 (2011), 39–67.

5.Dabbish, L., Stuart, H.C.S., Tsay, J., & Herbsleb, J. Social coding in GitHub: Transparency and collaboration in an open software repository. In *Proc. CSCW,* (2012), 1277-1286.

6.Derthick, K., Tsao, P., Kriplean, T., Borning, A.,Zachry, M., & McDonald, D. W. Collaborative sensemaking during admin permission granting in Wikipedia. In *Online Communities and Social Computing* (Vol. 6778), (2011), 100-109.

7.DiMicco, J., & Millen, D. Identity management: multiple presentations of self in facebook. *Proc. GROUP*, (2007), 383–386.

8.Donath, J. S. Identity and deception in the virtual community. *Communities in cyberspace*, (1996), 29–59.

9.Donath, J. Signals in social supernets. *Journal of Computer-Mediated Communication*, *13*(1), (2007), 231–251.

10.Ellison, N., Heino, R., & Gibbs, J. Managing Impressions Online: Self-Presentation Processes in the Online Dating Environment. *Journal of Computer-Mediated Communication*, *11*(2), (2006), 415–441.

11.Ellison, N. B., Hancock, J. T., & Toma, C. L. Profile as promise: A framework for conceptualizing veracity in online dating self-presentations. *New Media &Society*, *14*(1), (2012), 45–62.

12.Goffman, E. The presentation of self in everyday life. New York: Anchor Books, 1959.

13.Grasz, J. "One-in-Five Employers Use Social Networking Sites to Research Job Candidates, CareerBuilder.com survey finds."http://www.careerbuilder.com/share/aboutus/pressreleasesdetail.aspx?id=pr459&sd=9%2F10%2F2008&ed=12%2F31%2F2008 (Retrieved Feb 19, 2012)

14.Kristof-Brown, A., Barrick, M.R., and Franke, M. Applicant impression management: Dispositional influences and consequences for recruiter perceptions of fit and similarity. *Journal of Management 28*, 1, (2002), 27–46.

15.Lampe, C., Ellison, N., & Steinfield, C. A familiar face (book): profile elements as signals in an online social network. *Proc. CHI*, (2007), 435–444.

16.Lerner, J., Tirole, J., Some simple economics of open source. *Journal of Industrial Economics* 50 (2), (2002), 197–234.

17.Long, J. Open Source Software Development Experiences on the Students' Resumes: Do They Count? Insights from the Employers' Perspectives. *Journal of Information Technology Education*, 8, (2009), 229–242.

18.Mavlanova, T., Benbunan-Fich, R., and Koufaris, M. Signaling theory and information asymmetry in online commerce. *Information & Management*, (2012), 240-247.

19.Oreg, S. and Nov, O. Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in Human Behavior 24*, 5 (2008), 2055–2073.

20.Paul, S.A., Hong, L., and Chi, E.H. Who is Authoritative? Understanding Reputation Mechanisms in Quora. *Proc. Collective Intelligence,* (2012), *arXiv:1204.3724v1*.

21.http://programmers.stackexchange.com/questions/12070/would-you-put-your-stackoverflow-profile-link-on-your-cv-resume (Retrieved Aug 25, 2012).

22.Spence, M. Job market signaling. *The quarterly journal of Economics*, *87*(3), (1973), 355–374.

23.Spinellis, D. Open source and professional advancement. *Software, IEEE*, 23(5), (2006), 70–71.

24.Stewart, K.J. and Gosain, S. The Impact of Ideology on Effectiveness in Open Source Software Development Teams. *MIS Quarterly 30*(2), (2006), 291-314.

25.Tong, S.T., Van Der Heide, B., Langwell, L., and Walther, J.B. Too much of a good thing? The relationship between number of friends and interpersonal impressions on Facebook. *Journal of Computer-Mediated Communication 13*, 3 (2008), 531–549.

26.Trammell, K., & Keshelashvili, A. Examining the new influencers: A self-presentation study of A-list blogs. *Journalism and Mass Communication Quarterly*, 82(4), (2005), 968-982.

27.Walther, J.B., Van Der Heide, B., Hamel, L.M., and Shulman, H.C. Self-Generated Versus Other-Generated Statements and Impressions in Computer-Mediated Communication A Test of Warranting Theory Using Facebook. *Communication Research 36*(2), (2009), 229–253.

28.Weiss, B. and Feldman, R.S. Looking good and lying to do it: Deception as an impression management strategy in job interviews. *Journal of Applied Social Psychology 36*, 4 (2006), 1070–1086.