Social Media in Transparent Work Environments

Jason Tsay, Laura Dabbish, James Herbsleb
School of Computer Science and Center for the Future of Work, Heinz College
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
{jtsay, dabbish, jdh}@cs.cmu.edu

***Abstract*—Social media is being integrated into work environments making them more transparent. When the work environment is transparent, it has the potential to allow projects to transmit information about work artifacts and events quickly through a large network. Using signaling theory, we propose a theory that users interpret this information and then make work-related decisions about attention and effort allocation in a principled manner. In our research setting, an open source context of voluntary participation, broadcast activity information act as signals that allow developers to make highly informed choices about where to expend their attention and effort and with whom to collaborate. We propose four potential signals from literature and interviews with developers in our research setting and discuss the implications for social media in software development environments.**

***Index Terms*—Social media, open source, software development, social computing, signaling theory, transparency**

I. Introduction

Explosively popular social media in today's world enable users to create new ties, maintain relationships, and facilitate discussion with people all over the world [1]. The pervasive and widespread use of social networking sites such as Facebook and Twitter have not escaped the notice of both public and private organizations. Much like email and instant messaging before, enterprises, non-profits and even governments are increasingly trying to find ways to leverage social media to accomplish work [2].

Social media refers to internet-based applications that allow the creation and exchange of user generated content such as status updates, blog posts, and wiki articles [3]. In particular, we are interested in social media websites that allow individuals to articulate an interest network of people and artifacts, and receive updates about actions by those people or changes to those artifacts. These sites let users share a wide range of information through personal profiles, and provide a setting for interactions among individuals across physical and social boundaries [4]. The social network of participants is articulated through "friending" relationships, which enable participants to receive updates about their friends' activities in a feed. Despite the popularity for personal use [1] and application in a few corporate settings [5], there is little research about social media use in online work contexts.

Social media should strongly influence the way work is done, particularly in commons-based peer production communities. As Benkler [6] points out, commons-based peer production poses a significant information problem for participants, i.e., deciding where to apply their effort. Lacking in traditional resource allocation mechanisms such as markets and hierarchies, participants must self-coordinate, and projects must compete for attention and contribution.

In a typical commons-based peer production environment, there are a number of loosely connected projects that produce artifacts. There are also participants, who may freely contribute to project artifacts, discuss the project and its artifacts, and use artifacts for various purposes. The project artifacts and discussions are often publically visible to all members of the environment; open for any participant to decide to contribute their attention or effort towards the project. Transparent work environments such as these, when combined with social media, allow much information about work artifacts and events to be propagated over a broad network to any interested participant. However, there is much we still do not understand about how participants use this information to make work-related decisions. In particular, we are interested in how participants choose to allocate their attention or effort in a commons-based peer production environment where project membership is often fluid and voluntary. We address the following research question in order to advance our understanding of the relationship between work information broadcast social media and completing tasks:

How does work information broadcast by a project over social media influence attention and allocation of effort to this project?

To address this question, we propose that the information broadcast by projects in transparent work environments equipped with social media is received and interpreted by users in a principled manner. We draw from signaling theory to develop a theory of how the transfer and interpretation of work-related information that is broadcast over a social network takes place. We argue that activity information acts as a signal of internal project characteristics. These signals indicate projects that are more or less worthy of attention or contribution from members of the larger community. Guided by literature and interview data, we propose potential signals that projects in our research setting broadcast to potential contributors and hypothesize how they might influence how participants contribute to projects. The theory development we

present here is speculative; it is partially inspired by the data rather than validated by it. We do not attempt to make Empirical evaluation is something we hope to accomplish in the future, and we invite other interested researchers to join in that enterprise.

II. Social Media and Signaling Theory

In order to understand the impact of social media in work settings, we need to understand how information about projects and their members is interpreted and acted upon. We view this process through the lens of signaling theory [7], [8]. Signals are observable pieces of information that indicate some hidden quality of the person or entity that generated the signal [8]. Social media provides a rich new source of signals, potentially helping users decide how to distribute their attention and effort. Signaling theory is a useful lens for understanding how information broadcast in social media could affect the distribution of attention and contribution in online settings. Transparent work environments provide a venue where interactions and actions are broadcast over social media and can act as signals.

Signals are observable pieces of information that indicate hidden qualities of the signaler [8]. For example, a fur coat can signal the wearer is wealthy. Some signals are inherently reliable, but others may be deceptive and indicate qualities that the signaler does not actually possess. “Assessment” signals are reliable signals where the indicated quality is inherent to the signal. Writing correct code for complex functionality is a reliable signal for coding skill, for example. “Conventional” signals, on the other hand, are potentially unreliable signals that indicate a quality simply by agreement among members of a community. For example, a resume listing involvement in a number of impressive-sounding projects may or may not reflect meaningful experience and skill. Donath [7] finds rich patterns of signaling and deception in online communities to infer member identities.

In a transparent work environment equipped with social media, work activities for projects are not only public but are broadcast over the social network for interested parties to observe. In contrast to mainstream social networking services popular today, work artifacts and events are treated as first-class objects in the network of this kind of social media. For example, an entire network of users depending on a certain software library can be notified whenever an important feature is added. The work events generated by project members performing tasks on work artifacts are broadcast over the project's social network. A project's broadcast work events may act as signals for users to interpret. Users interpret the signals to infer different project qualities such as quality, collaborative environment, and member commitment. These users then make work-related decisions based on the interpretation of the project's signals. This process may be especially important in a commons-based peer production system like most open source projects as the work and coordinated efforts are often voluntary [9]. Open source projects often must attract both attention and contributions from the community in order to survive and thrive. Users interpreting signals from projects often then decide whether or not to contribute to the project based on project qualities determined from the signal.

III. Research Setting

To guide our theory development, we drew upon a set of semi-structured interviews of GitHub developers. These interviews are described in more detail in prior work [10].

For this paper, we use these interviews as a source of examples to help develop our theoretical contributions applying signaling theory to open source ecosystems.

To investigate how work is done in a transparent environment equipped with social media, we chose the GitHub software project hosting service as our research setting. GitHub implements many of the social networking features found in well-known social networking sites such as Facebook and Twitter to improve collaboration between software developers. These features include the ability for developers to "follow" other members in the community (in the same or other projects) and to "watch" the repositories of different projects. Watching and following direct events about changes to a repository or actions by a developer to the participant’s news feed. Examples of such events include commits (changes to the repository code base) on a watched repository, comments made by fellow contributors on those commits, or the creation of a new project by a followed participant. Much of the followed participants’ social activity is also visible in the feed, including changes to the set of users that person is following. In addition, participants have a profile page that lists personal information as well as activity-related information such as the repositories they own and watch as well as the participants that they follow..

Our interviews suggested that the following signals were central influences on work-related actions in a transparent environment: high status participants, relative commitment, the existence of a clear core development team, and the existence of a clear core communication team. In the following sections, we draw from literature and interviews to develop hypotheses about each signal’s influence on users’ contributions of attention or effort to a project.

IV. High Status Participants

Attention is a scarce commodity in a social media environment, and users who get a lot of attention have higher status in these settings [10], [11]. This level of attention is easily accessible as a count of the number of people subscribed to an individual’s activity in the interest network. A user who chooses to follow another participant, i.e. add that participant to the user’s interest network, publicly subscribes to updates on the participant’s activities. In a transparent environment where the online profile of the developers who participate in a project is not only public but broadcast over a large social work, users may observe as a signal that these high-status developers are participating in a particular project. Users may then use this information to make decisions on whether or not to distribute their attention or effort to this project.

A. Interview Data

We found in prior work on GitHub [10] that the number of followers is a signal of status in the GitHub developer community. It may function as an assessment signal because it is difficult to falsify. For a participant to have a high follower count, many other participants must have acted on a decision to publicly follow this particular participant.

In interviews with GitHub developers, we found that developers made use of the following relation and user reputation to help make decisions on what projects they should attend to. Users looking for interesting projects would look for projects that popular developers associated themselves with. As one developer put it:

"[very well-known developer], anything that he commits to is probably worth watching. He does a lot of really interesting work so I follow him for the interest level. Same thing with [other very well-known developer]."

In this environment, where attention is a scarce commodity, users make use of popular developers as a signal of what projects they should contribute their attention to. Prior work on GitHub [10] also finds that projects with higher measures of attention, which are publically visible in this transparent work environment, tend to be judged as higher quality by members of the community. Users in our research setting were also aware that the number of followers conferred a sense of social capital:

"...there's a certain capital to [following], you're following someone and increasing their number of followers and that was my main reason for following people."

Similar to the case of having too many Facebook friends, we found evidence that having too many followers or certain patterns of follower relationships actually signaled dishonesty rather than status:

"A few of them are GitHub follow spammers. I can tell because I go to their profiles and there's nothing but they follow like 150, 200 people but they don't have anything."

In signaling theory terms, this may be an example of the community imposing norms to keep this signal reliable by imposing an informal punishment cost [8] on dishonesty. In both cases, the punishment is that there is a negative interpretation on the sender's desired quality (status or social attractiveness) by the receiver.

B. Literature

Related work on transparent work environments and social media suggests that high status users may be more influential than others. Kraut and Resnick [12] make the claim that requests from high status participants in online communities draw more contributions than anonymous or less visible participants. These high status participants may hold authority, such as a moderator in an online forum. This status may also occur from participating in the community in a particularly visible way, such as being a frequent editor in a Wikipedia article. In social media, Preussler and Kerres [11] find a similar pattern in the microblogging service Twitter, where they find a link between social reputation on the service and the number of followers a user has. Kwak et al. [13] find that Twitter users are ranked similarly when comparing ranking by number of followers to ranking using a PageRank algorithm that measures relative importance using a model for influence [14]. Tong et al. [15] find in the social networking service Facebook that users with an above-average number of friends are associated with higher ratings of both social attractiveness and extraversion. At the same time, Facebook users with an excessively high number of friends then were associated with lower ratings of social attractiveness and extraversion. In signaling terms, an excessively high number of friends actually works more like a conventional signal in that receivers now potentially suspect the number of friends to be artificially inflated by some means. The signal is no longer reliable but societal norms appear to be in place to punish dishonesty. This parallels the "follow spammer" case seen in the interviews.

C. Measure

In our research setting, developers may publically subscribe to the events generated by other developers through the "follow" relationship. Examples of events generated by following developers include notifications of when they create new projects or join existing projects. Based on prior work, we use the number of followers that a developer has as a signal of the developer's status within the community. As the profiles of developers on GitHub are public, users looking at a project may notice that the developers associated with the project have a higher-than-average number of followers. They interpret this as a signal of status and make decisions influenced by this interpretation.

V. Relative Commitment

In a commons-based peer production work environment, participants volunteer their effort to any number of projects as they see fit. Some may choose to focus their attention on a single task while others may find it necessary to participate in a large number of projects to achieve some larger goal. In a transparent environment equipped with social media where the public work actions of a project's members are broadcast over a social network, participants are able to see whether the developers on a project choose to focus disproportionately on that project or tend to spread their contributions out somewhat equally over many projects. These participants may then interpret this signal as a developer's relative commitment to a particular project and use this information to decide whether or not the project is worth volunteering their attention or effort towards.

A. Interview Data

Prior work on GitHub [10] has found that members in the research setting use the recency and volume of activity performed by developers in particular projects as signals of the

interest and level of commitment by the developers for the project. Many of the projects on the GitHub service are either abandoned or not well-maintained. Participants looking for useful or interesting projects make use of this signal in order to determine whether or not a project is worth the attention or effort. In interviews with GitHub developers, we found that some participants, when trying to select between two similar projects, would use signals of relative commitment of projects' developers to determine which project is more appropriate:

"Well there's this one-- there's this project called [Project A], that I've been really trying to see how good it is, versus [Project B], and I think that maybe early on… [Project B] had-- was better at first, but then [Project A], this guy on [Project B] is just-- a machine, he just keeps cranking out code… people are invested in this project"

In this scenario, the observing participant selects the project that broadcasts a signal of having developers with more commitment towards the project.

B. Literature

Literature on open source software and online communities finds that commitment to a project tends to draw more contributions. Kraut and Resnick [12] make the claim that commitment to a particular online community, both as a whole and to particular people, increases the willingness to contribute to it. At the same time, they make the claim that participants are more likely to comply with requests when they see that other participants have complied. One possible explanation offered for this decision is that observing commitment gives social proof [16] that contributing to this particular project is appropriate. In terms of signaling theory, users are able to observe signals of commitment from project participants and may be more likely to contribute. Lakhani and Hippel [17] in examining the motivations for open source software participants to provide voluntary field support, found that many of the members would provide support in order to "help the cause". The members in question strongly identified themselves as part of a community and were more likely to provide assistance to others. The same survey also indicates that members are highly motivated by reciprocation, if they had received help, they would be more likely to then give help. The same work also drew parallels to the work of Constant et al. [18] on the motivations of an industry help line. They found that the information providers rated "being a good company citizen" as the most important reason for replying. We see that in both open source communities and industry, being committed seems to encourage responses and contribution.

C. Measure

In our research setting, the work actions of developers are both public and broadcast over a social network. Users are able to observe the distribution of work that a developer performs across projects, at least in the recent history that is captured in the newsfeed and project status graphics. As with most peer-production communities, developers vary widely in overall time and effort they can contribute to open source [19]. For this reason, the absolute level of contribution to a project may be less meaningful than a developer's relative contribution. For example, 4 commits a week on Project A may represent a strong "vote" for the importance of that project from a developer that contributes only 5 commits a week overall. On the other hand, 4 commits a week on Project A may not be a "vote" for Project A at all for a developer that averages 50 commits a week, 40 of which go to Project B. We can measure this relative contribution towards a certain project for a particular developer by looking at the proportion of a developer's total contributions that are allocated towards that particular project. To measure the signal of relative commitment on the project level, we can use the distribution of the developers' work over projects across all of the developers in the project. For example, we could use the mean of the proportions of each developer's commits devoted to this project, perhaps weighted by the number of a developer's commits to this project. As participants must actually perform work in one or more projects to generate this signal, this is an inherently credible assessment signal of how effort is allocated. Users looking to contribute to a project may be more willing to contribute if they observe that the project's developers contribute their effort disproportionately to this particular project.

VI. Existence of Clear Core Development Team

Work highly concentrated in a few participants signals the existence of a relatively small "core" insider team. Many highly successful open source software projects [20] as well as other well-known crowdsourcing efforts such as Wikipedia [21] have highly non-uniform distributions of work. In the previous section, we were looking at the way developers allocated their effort over projects, as a measure of one dimension of commitment. Here we are looking at how the total effort for a single project is distributed over developers. When there is a core team that assumed a large share of the development work, one would expect to see this reflected in the distribution of effort over individuals in the project. In a transparent environment equipped with social media, the distribution and authorship of work performed in a project is broadcast over the social network. Interested participants are able to observe these broadcast work events and determine whether or not a small group of people are authoring the majority of the project's work and can readily see the identities of these core developers.

A. Interview Data

Prior work on GitHub [10] finds that developers often read a project's record of work activity in order to make a number of inferences. Developers would use this record to not only determine how the project has evolved but also infer the overall structure of the project and its members. As part of determining the structure of the project, these developers would also determine who were the core members and which particular parts of the project each member had expertise in. In interviews with GitHub developers, we found that participants in projects would use the social media in the service in order

to keep track of the activities of their fellow project members. One particular developer also notes the relative volume of activity that the fellow core member is producing:

"...this is kind of the first [project] where [other core member] is committing a lot more than I am, so I get to keep up with what he's doing and read the [...] commits, all that fun stuff."

In signaling theory terms, the respondent developer has observed the relative difference in the distribution of work on this project and perhaps has interpreted this as a signal of relative visibility as a core member of this project. Noting that the fellow core member seems to be visibly making many more changes, the respondent has made the decision to pay more attention to the core member's activity.

B. Literature

Related work on open source software and online production communities suggests that a visible core team motivates work contributions to the group. Lerner and Tirole [22] observe that leaders in open source projects tend to be developers who author the initial code base or create important early contributions. These leaders tend to possess not "formal authority" (cannot force decisions onto members) but "real authority" (recommendations are respected and followed by members). In this way, signals for the existence of a core team also signal the presence of leadership and this "real authority". Other online production communities also find similar distributions of work. Ortega et al. [21] found that the majority of Wikipedia articles had very unequal distributions of work, where small numbers of editors make the majority of contributions. Kittur and Kraut [23] find in their analysis of Wikipedia articles that when work is concentrated among a few editors, the quality of Wikipedia articles was higher. They interpreted this result as an indicator that high concentration of effort across individuals was a form of implicit coordination, reducing process overhead and the need for explicit coordination through communication media. Coordination overhead is a cost that a potential developer must pay in order to contribute to a project, and the lower the difficulty or cost of participation, the more active people are likely to be.

In online communities, Kraut and Resnick [12] make the claim that normative commitment, the feeling that one has obligations to a community, is increased in a community when testimonials of other members' normative commitments are visible. They also review evidence suggesting that participants in a group are more willing to contribute when the group is small rather than large. Lastly, participants are more willing to contribute when they think that their contributes are unique and not interchangeable with other group members. These three claims combined suggest that within a project, members of the core team are motivated to contribute when there exists a small core group making highly visible contributions and where each member's contributions are essential. Using Karau and William's [24] collective effort model, core members of a project may feel obligated to contribute because they realize that their contributions are unique to the group and have a direct impact on the success of the project.

C. Measure

The visible record of project activity signals how work is accomplished inside of a project. As all of the work performed in a project is made visible, participants can readily see the relative concentration of work per project member. For example, a project may have each member making fairly equal contributions while another project may have one or two members authoring the majority of artifact content. One could use this concentration of work as a signal of the extent to which a project has a visible core development team. One measure for this concentration of work is the Gini coefficient [25], which ranges from 0 to 1. A value of 0 represents a project where work is performed evenly across all members, and a value of 1 represents a project where work is performed solely by a single member. For projects where a few members author the majority of the work in a project, that signals a clear core team whereas a project with work spread evenly over many members would signal a much less visible core team. As participants must actually perform work in the particular project to generate this signal, this is an inherently credible assessment signal of the distribution of work. Participants in a project can look at the project's record of activity and use the signal of the core development team to motivate their own contributions.

VII. Existence of Clear Core Communication Team

One distinctive feature of a work environment supported by social media is that discussions about work artifacts are tied directly to the artifacts, and these discussions are broadcast to the network of those who have chosen to receive them. From this, users can also see the distribution of the authorship of such comments. For example, one or two members may author a majority of comments in a project, or they may come more equally from a larger group. Just as it was the case for a core development team (Section VI), it seems likely that when a small number of people contribute most comments, it signals the existence of "core" communicators in the project. These individual members' styles, opinions, and expertise are readily discoverable because of repeated exposure. It also indicates the likelihood that if an outsider interacts with a project, any response they get is likely to come from a member of the core.

A. Interview Data

Prior work on GitHub [10] shows that developers use the discussion system on work artifacts in a number of ways. When a user contributes to a project, often a member of the project acts as a core communicator and starts a discussion over the contribution. Oftentimes this discussion is feedback for the contribution, with the core communicator offering comments on style, correctness, or efficiency. In other cases, the discussion around the contribution resembled more of a negotiation as the contribution might conflict with a core member's "vision" for the trajectory for the project. There were also cases where a change with the potential to conflict with a dependent project resulted in a core member from the dependent project participating in the discussion. In other cases, representatives from organizations or companies who

depended on the project might also join the discussion for similar reasons. Some users used the existence of discussion as a signal for how interesting a work artifact or event would be. For example, a change with a lot of comments seems more interesting and worth investigating compared to a quiet change with no discussion. Some developers would initiate discussions in order to learn more about the project and how to build upon the project. Some core communicators in particular would often be very forthcoming with this type of help, even using communication channels outside of the research setting, such as Twitter or Internet Relay Chat (IRC). For some core members, these developers represented opportunities to nurture a potential contributor. In interviews with GitHub developers, we also found that core members found it useful to have other core members start discussions on contributions:

"I'm really trying to get [other core members] to comment more on my code, because I find it's a good way to do really informal code review... But this is a way, you know, if you check in code, somebody else will have their eyes on it and if there's an issue or something, you can do a little bit more efficiently they can just leave a note."

Core communicators also tended to pay more attention to the comments that other core members made, perhaps as a result of reciprocity. We also found cases where core communicators of popular projects would become overwhelmed by having to respond to communication requests by other users.

B. Literature

Related work on social media and open source software finds that a small, dedicated group of communicators for a project tends to draw contributions from outside of the group. Similar to the case with core developers, combining the claim of Kraut and Resnick [12] that participants in online communities are more willing to contribute when they think that their contributions are unique with the collective effort model [24], the core members of a project may feel obligated to respond to communications that non-core members are not able to answer, such as a question requiring the expertise of a core member of the project. Kraut and Resnick [12] also present evidence suggesting that users are more likely to respond to requests the more they like the requester or the more familiar the requester is to the user. Repeated exposure to a small group of core communicators allows for users to become more familiar with these representatives of the project. Von Krogh et al. [26] found a similar case in open source software, where potential joiners to a project would often spend anywhere from several weeks to several months observing the project mailing list before contributing. Before making the decision to contribute to the project's mailing list, potential members would spend the time to become familiar with the existing project participants and the culture of the project itself.

C. Measure

As the discussions over work artifacts and events are public and tied directly to the artifacts, users are able to observe which project members tend to participate in discussions. In our research setting, each discussion actually generates its own work event that is propagated over the social network. One could use a measure of the non-uniformity of the distribution of communications over project members as a signal of whether or not a project has a visible core communication team. The Gini coefficient could also be used here to measure the inequality of the distribution of communication. For example, a project where only a single member authors all of the communication artifacts in the project would have a Gini coefficient of 1. In a project where the communications are spread evenly across all of its members, the Gini coefficient would be 0. As participants must actually author discussion artifacts in the particular project to generate this signal, this is an assessment signal of how communication is distributed. Participants or potential participants to a project are able to observe the discussions in a project and use the signal of the existence of a core communication team to predict the likely responses to a potential contribution.

VIII. Discussion

Our intended contribution in this paper is the development and application of theory, based on signaling theory, to open source ecosystems. We develop a number of conjectures that could be tested in future empirical work. Based on literature and a few examples taken from interviews with GitHub developers, we speculate about four potential signals that projects broadcast for users in their social network as well as potential measures for each signal. We summarize these signals and measures in Figure 1.

From each of these signals, we also develop hypotheses of how users who receive these signals interpret them and then act in relation to the broadcasting project. The hypotheses are as follows:

H1: Projects that signal the inclusion of High Status Participants are likely to have users contributing attention to the project.

Literature suggests that having a high status or reputation in terms of social media is associated with a high influence and visibility. Interviewed developers mention using high status users as a way to determine what projects are potentially interesting. Unlike in much of industrial software development [22], developers on popular open source projects are highly visible, and project membership is often public. Users receiving a signal that a project contains these highly visible developers should then be likely to contribute their attention towards this project.

H2: Projects that signal the inclusion of participants with a high Relative Commitment are likely to have external users making new contributions.

Literature suggests that visible commitment to a certain group tends to draw contributions. GitHub interviewees also valued projects where developers were more committed when

High Status Participants	Number of Followers	Users contribute more attention
Relative Commitment	Proportion of Developers' Commits to this Project	More new users make contributions
Existence of Clear Core Development Team	Gini coefficient for distribution of work	Internal users contribute more effort
Existence of Clear Core Communication Team	Gini coefficient for distribution of communication	More external users make contributions

Fig 1. Summary table of signals, measures for signals, and hypothesized effect on users for project outcomes.

deciding between multiple similar projects. Users making the decision whether or not to contribute to a project may use signals of high relative commitment as social proof that the project is appropriate to contribute to.

H3: Projects that signal the existence of a Clear Core Development Team tend to produce more contributions from within the project.

Interviews suggest that developers internal to a project observe fellow core project members and act upon signals broadcast by the project. Literature suggests that these signals may be motivating to core developers so long as these developers are contributing work that is irreplaceable by developers peripheral to the project. Participants internal to a project may use signals to determine their position as a core or peripheral developer in a project. Without the feedback that a core developer is doing work unique from peripheral developers, productivity may drop due to social loafing [24].

H4: Projects that signal the existence of a Clear Core Communication Team tend to draw more contributions external to the project.

Interviewed developers suggested that when core members of a project participate in discussions, they play a number of important roles such as giving useful feedback and providing opportunities to learn. From literature, as users become more familiar with core communicators, they may be more willing tp participate in discussions. The smaller the group of core communicators, the easier it is for users to familiarize themselves with this group through repeated exposure. Users external to a project may be more willing to contribute effort if they receive signals describing that core members of the project are willing to communicate and provide feedback or learning opportunities.

Future work will investigate these hypotheses as they apply to our research setting. Also, although we have identified a number of potential signals, there are undoubtedly many more signals that projects or communities propagate over the social network that participants can make use of. Should the hypotheses hold, there are also potential design implications that may arise for future collaborative software development environments. For the software development environments of the future that are both transparent and make use of social media, both projects and tools have opportunities to leverage the broadcasting of signals to achieve desired effects. For example, a future development environment may realize that the current user is a core member of a certain project. Then, depending on whether or not the current project needs more internal or external contributions, the development environment may choose to display different signals to this user such as information on what fellow core members are working on. Using signaling theory to further our understanding of how users interpret information broadcast over a social network could also help reduce information overload problems that may arise from using tools equipped with social media. Instead of receiving every single broadcast event like in traditional social media services, filters that only display events that signal qualities related to a user's task could be put in place.

IX. Conclusion

In this work, we proposed a preliminary theory for understanding how users in a transparent environment equipped with social media interpret and act upon work-related information that is broadcast by projects over a social network. We use signaling theory to describe the transfer and interpretation of information and how this information may affect the work-related decisions that users make in this environment. In our research setting of GitHub, we proposed four signals that open source software developers may make use of when deciding to contribute attention or effort to projects. These signals suggest hypotheses that may bring insight into how users in this kind of environment make decisions after interpreting certain signals. By understanding how to make use of information that is broadcast over social networks, we may inform the design of future development environments for large-scale collaboration and imply a variety of ways that transparency can support innovation, knowledge sharing, and community building.

Acknowledgment

This research is supported by the Center for the Future of Work at Carnegie Mellon University's Heinz College and by the National Science Foundation under Grant No. IIS1111750.

References

[1] danah m. boyd and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[2] M. M. Skeels and J. Grudin, "When social networks cross boundaries: a case study of workplace use of facebook and linkedin," in *Proceedings of the ACM 2009 international conference on Supporting group work*, 2009, pp. 95–104.

[3] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media," *Business Horizons*, vol. 53, no. 1, pp. 59–68, Jan. 2010.

[4] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng, "The impact of social media on software engineering

practices and tools," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, 2010, pp. 359–364.

[5] J. DiMicco, D. R. Millen, W. Geyer, C. Dugan, B. Brownholtz, and M. Muller, "Motivations for social networking at work," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008, pp. 711–720.

[6] Y. Benkler, "Coase's Penguin, or Linux and the Nature of the Firm," *The Yale Law Journal*, vol. 112, no. 3, pp. 369–446, 2002.

[7] J. Donath, "Signals in Social Supernets," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 231–251, 2007.

[8] J. Donath, *Signals, truth, and design*. Cambridge, MA: MIT Press, 2005.

[9] A. Bonaccorsi and C. Rossi, "Why Open Source software can succeed," *Research Policy*, vol. 32, no. 7, pp. 1243–1258, 2003.

[10] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in GitHub: transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 2012, pp. 1277–1286.

[11] A. Preussler and M. Kerres, "Managing reputation by generating followers on Twitter," in *Medien – Wissen – Bildung Explorationen visualisierter und kollaborativer Wissensräume*, 1st ed., T. Hug and R. Maier, Eds. Innsbruck: Univ. Press, 2010, pp. 129–143.

[12] R. E. Kraut and P. Resnick, *Building Successful Online Communities: Evidence-Based Social Design*. Cambridge, MA: MIT Press, 2012.

[13] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 591–600.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web.," Stanford InfoLab, Nov. 1999.

[15] J. B. Walther, B. Van Der Heide, S.-Y. Kim, D. Westerman, and S. T. Tong, "The Role of Friends' Appearance and Behavior on Evaluations of Individuals on Facebook: Are We Known by the Company We Keep?," *Human Communication Research*, vol. 34, no. 1, pp. 28–49, 2008.

[16] R. B. Cialdini and N. J. Goldstein, *Influence: Science and Practice*, 4th ed. New York: Allyn and Bacon, 2001.

[17] K. R. Lakhani and E. von Hippel, "How open source software works: 'free' user-to-user assistance," *Research Policy*, vol. 32, no. 6, pp. 923–943, 2003.

[18] D. Constant, L. Sproull, and S. Kiesler, "The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice," *Organization Science*, vol. 7, no. 2, pp. 119–135, 1996.

[19] R. A. Ghosh, "Understanding free software developers: Findings from the FLOSS study," *Perspectives on free and open source software*, pp. 23–46, 2005.

[20] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and Mozilla," *ACM Trans. Softw. Eng. Methodol.*, vol. 11, no. 3, pp. 309–346, Jul. 2002.

[21] F. Ortega, J. M. Gonzalez-Barahona, and G. Robles, "On the Inequality of Contributions to Wikipedia," in *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, 2008, pp. 304–.

[22] J. Lerner and J. Tirole, "Some Simple Economics of Open Source," *The Journal of Industrial Economics*, vol. 50, no. 2, pp. 197–234, 2002.

[23] A. Kittur and R. E. Kraut, "Harnessing the wisdom of crowds in wikipedia: quality through coordination," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008, pp. 37–46.

[24] S. J. Karau and K. D. Williams, "Social Loafing: A Meta-Analytic Review and Theoretical Integration," *Journal of Personality and Social Psychology*, vol. 65, no. 4, pp. 681–706, Oct. 1993.

[25] D. A. Harrison and K. J. Klein, "What's the difference? Diversity constructs as separation, variety, or disparity in organizations," *Academy of Management Review*, vol. 32, no. 4, pp. 1199–1228, 2007.

[26] G. von Krogh, S. Spaeth, and K. R. Lakhani, "Community, joining, and specialization in open source software innovation: a case study," *Research Policy*, vol. 32, no. 7, pp. 1217–1241, 2003.