What is Good Feedback in Big Data Projects for Cyberinfrastructure Diffusion in e-Science?

Kerk F. Kee
School of Communication
Chapman University
Orange, CA, USA
kee@chapman.edu

Jamie C. McCain
School of Communication
Chapman University
Orange, CA, USA
mccai101@mail.chapman.edu

***Abstract*—This paper investigates the role of feedback in big data projects for cyberinfrastructure (CI) diffusion in e-science. For many of these projects, large-scale and heterogeneous datasets, multidisciplinary and dispersed experts, and advanced technologies are brought together to harness analytic insights. However, without effective CI and computational tools, the accuracy and meaningfulness of analytics results are compromised. In fact, without CI tools, raw data remain raw with hidden insights, as data analytics cannot be executed at all. In order to improve such tools for meaningful results, we argue to conceptualize the communication mechanism of 'feedback' in agile software development, with the goal of producing CI tools that are responsive to users. Based on a grounded analysis of interview data, we concluded that feedback helps developers in big data projects understand users' needs, makes tools user-friendly, prevents emergencies, and is better for developers than no feedback. Furthermore, good feedback is often structured, specific, actionable, timely, generalizable, and delivered in a tactful way. Despite the limitation of the findings being exploratory and yet to be evaluated experimentally, we argued that they still can motivate developers to be proactive seekers of feedback for their tools, productively guide developers' communication with users, and ultimately promote further adoption and diffusion of CI tools in e-science.**

***Keywords*—feedback, agile software development, e-science, cyberinfrastructure, technology adoption, diffusion of innovations**

I. Introduction

Big data projects are complex socio-technical phenomenon. These projects often involve large-scale and heterogeneous datasets, multidisciplinary and oftentimes dispersed teams of experts, and the development and use of advanced technologies, in order to harness big data for insights. In some projects, the technological tools are designed with in-house users in mind, while others are designed for a broad and general user community. In either case, users are seldom integrally involved throughout the technology development process, especially when they come to the technologies at later stages of adoption and diffusion. The challenges of irregular user involvement in the early stages and the adoption experience of users in the later stages can lead to limited feedback useful for developers in big data projects. Furthermore, much feedback is communicated ineffectively, leading to unproductive communication for technology improvement. Without effective technologies, the accuracy and meaningfulness of analytics results are compromised. In fact, without analytics technologies, raw data remain raw with hidden insights, despite the big scale of the dataset, as data analytics cannot be executed at all. In other words, big data projects face a critical challenge of good feedback communication with users in order to develop effective technologies.

The aforementioned challenges are especially true in the realm of e-science, where scientists in different domains, such as high energy physics, computational chemistry, and bioinformatics, increasingly rely on new computational tools to perform large-scale and data-intensive research in science. For most e-science projects, or big data projects in various scientific fields, the analytics technologies are often developed out of research projects federally funded by the National Science Foundation, National Institutes of Health, Department of Defense, Department of Energy, etc. Because the funding is usually short-term, the developers behind these technologies in e-science, or cyberinfrastructure (CI) [1], make the tools open source, and push them out to domain communities, in order to attract new users, new developers, and scientist-developers (scientists who can code tools for scientific research) to carry the tools forward for the long-term [2]. In order for the tools to survive when the funding ends, and remain sustainable beyond the inception projects, being responsive to users in order to promote further adoption and systemic diffusion is key. However, the communication challenge of user feedback is further compounded when the projects are running out of funding, and the user-base at the later stages of adoption and diffusion is much more diverse than in the early stages of design and development.

This paper investigates the communication mechanism of feedback between users and developers in the context of big data projects that develop CI for e-science in the western world, primarily the US. In order to achieve this goal, we seek to better understand and articulate the rationales for why feedback is important in these big data projects. Furthermore, we explore in the context of e-science, what makes a piece of user feedback regarded as 'good feedback' for big data projects to build CI. We investigate these two research foci, because if developers are more proactive in seeking feedback, and feedback communication can be better conceptually defined, we argue that more productive feedback communication can be facilitated between users and developers in big data projects.

This material is based upon work supported by the US National Science Foundation (NSF) under award ACI-1322305.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-1-5386-5035-6/18/$31.00 ©2018 IEEE

This can lead to better CI tools to enable more accurate and meaningful analytics results, and further CI diffusion in various scientific domain communities.

This paper contributes to research on methodologies to improve data projects in three ways. First, it provides a deeper understanding of how feedback communication can improve data project outcomes, such as developing effective CI tools and promoting their subsequent adoption and diffusion. Second, it helps us better understand the complex role of feedback in agile software development, making users and user feedback more integral to the technology development process. Third, understanding good feedback communication helps improve software engineering project management. In order to explore the stated research foci, the paper is organized in the following fashion. First, we briefly review the literature on feedback. Second, we describe our data collection and analysis processes to investigate the research foci. Third, we present our findings. Fourth, we discuss the limitations of our research study. Finally, we wrap up the paper with a conclusion and several implications. The next section continues with a literature review on feedback.

II. RESEARCH LITERATURE ON FEEDBACK

A. *Feedback as Communication Loops in Mathematical Transactional Model*

The theorizing of feedback as a communication process could be traced to Shannon and Weaver, in their Mathematical Theory of Communication [3]. Shannon and Weaver are mathematicians and electrical engineers who were interested in modeling human communication. Their model is often explained in basic communication textbooks in a schematic. The schematic (such as the one in Figure 1) illustrates communication between an information source and a receiver. The information source (who encodes) sends a message through a channel to the receiver (who decodes). The model also includes other components such as noise which interrupts the message being sent to the receiver. Furthermore, Fiske explained that although it is not explicitly elaborated in Shannon and Weaver's model, feedback makes the process of message transmission and human communication more efficient [4]. In this way, feedback can be understood as the receiver's communicative reaction back to the sender, forming a loop from the destination back to the information source [5].

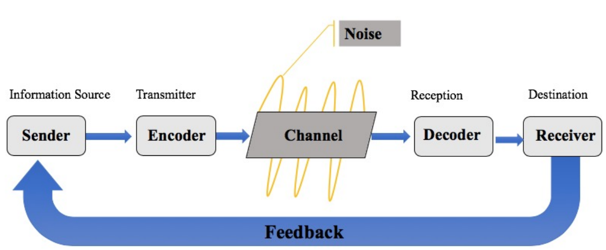

Fig. 1. Shannon and Weaver's Mathematical Model of Communication

B. *Feedback Improves Performance in Organization*

In the fields of organizational science and communication, feedback has been studied as a communication mechanism to help workers improve job performance. Researchers in this area argued that feedback is important for improving organizational outcomes. For example, Ilgen, Fisher, and Taylor [6] advanced the understanding that feedback message should come from a credible and valuable source, especially from someone who is viewed as a trustworthy expert by others within the workplace. Credibility of a feedback source is likely to induce a positive response from a feedback receiver. The receiver then responds to the feedback message based on perception of the feedback received (usually shaped by psychological closeness with source), acceptance of the feedback message (often influenced by belief that the feedback is accurate), desire to respond to the feedback given, and finally the intention to change. However, the ultimate behavioral changes also largely depend on the work environment, if there are major constraints and if the changes are within the receiver's control.

Subsequently, Taylor, Fisher, and Ilgen [7] expanded on this earlier understanding to include control theory, which maintains that receiver responds to feedback with the following possibilities: (a) accept feedback and make behavioral changes, (b) disagree with the source or performance standard, thus discounting feedback, (c) simply ignore the feedback, or (d) change the performance standard. Their work emphasizes the feedback characteristics and feedback processes on the behaviors of the receiver. In addition to source credibility discussed earlier, they argued that the characteristics of feedback message play a key role in the receptivity of feedback. In other words, these organizational researchers theorized that the positivity or valence of the feedback message impacts the receiver's behavior and reaction to the feedback itself [6; 8]. However, Taylor and colleagues did not elaborate on how the feedback message can be or should be communicatively designed for optimal impacts, an opportunity this paper pursues.

Building upon the early work on performance in organizations, London and Smither's research focuses on feedback as an important driver for performance management [9]. Specifically, individuals' orientation towards feedback shapes the organizational culture, and therefore sets the tone for how feedback is interpreted, used, and perceived in the performance management process of an organization. This process creates an iterative cycle of communication that the receiver can then understand and make an active choice about the given performance information. Arguably, creating a positive culture of feedback heavily influences the organization's performance as a whole. In other words, the context of an organization matters in how feedback is received. In the context of big data projects in e-science, a positive culture of feedback can strongly promote better development of CI tools for research analytics in the sciences.

In order to further understand positive culture of feedback, Moss, Valenzi, and Taggart explained that feedback seeking is rooted in impression management where an individual manages their interactions and impressions with superiors [10].

However, feedback seeking has evolved where individuals can reduce their uncertainty by engaging in active strategies such as requesting feedback from their coworkers or superiors than waiting for feedback [11]. Importantly, the organizational culture must support feedback elicitation and development towards seeking feedback [6]. Also, a study by van der Rijt, van den Bossche, and Segers showed that depending on the individual's position in the department affects seeking feedback, and that employees prefer to seek out feedback in their department across multiple levels [12]. Feedback seeking has been examined in organizational research over the years for impression management, specifically on performance within the workplace. In the context of big data projects in e-science, we argue that feedback seeking for impression management purposes is important for strategy with users for continuously promoting the adoption and diffusion of CI tools.

In the contemporary society, organizations have expanded to include various forms of dispersed teams and virtual organizations. In a study by Walther, Liang, and colleagues, feedback received via computer-mediated-communication (CMC) was found to be particularly persuasive in influencing the modification of a person's behavior [13]. Feedback in dispersed teams and virtual organizations, especially big data projects in e-science, should be further studied to understand the motivations and rationales behind feedback seeking, and how feedback can improve project's performance. We argue that feedback should be received through various channels, including through CMC channels, in order to cater to the diverse communication preferences of experienced and new users.

C. *Feedback is Key in Agile Software Development*

More importantly for big data projects, feedback is a key communication process in agile software development. The methodology of agile software development is powerful because it overcomes the limitations of the waterfall methodology by including frequent and constant user feedback in the technology development process. In other words, feedback is what makes agile software development a different and arguably more effective software development method than waterfall. This is not surprising given the arguments presented earlier that feedback provides a receiver's reaction to the information source, and feedback communicates certain implied performance standards and job expectations for workers.

For big data projects, feedback is critical in the development process due to the dynamic nature of requirements as well as given information about the cost and benefits of development activities within the environment [14]. Importantly, feedback process not only generates more information, it also increases learning and provides a clearer direction for software development [14]. Feedback is vital in the development process of big data projects, including those in e-science.

Big data projects consist of teams who employ critical success factors to execute the projects. Saltz, Yilmazel, and Yilmazel's study revealed that one of the challenges for software engineers is the lack of focus on key roles required to execute a big data project [15]. Importantly, Saltz and colleagues highlighted in their study that the important skill that data engineers need, which is often overlooked, is the communication skill to creatively solve problems. We argue that feedback communication, including seeking and managing feedback, is part of the communication skill essential for the success of big data projects.

Although agile software development practices can sometimes cause dismay within teams, due to the relocation of authority and accountability, these practices greatly improve the quality of the software products developed [16]. Moreover, Lycett, Macredie, and Paul elaborated on the value of the agile methodology, in that management can determine whether software development work is necessary and sufficient at various points during the process. They recommended using these practices for collaboration, interaction, and communication [17]. Similarly, Saltz and Shamshurin's case study presents a process flow which highlighted communicating at each step in the process. Specifically, communication is critical during the analysis process to convey preliminary results in analysis and dissemination steps, which is usually executed by the data science team [18]. Big data project teams communicate feedback in each iteration because feedback is integral to the agile software development methodology.

In addition to building team members who have excellent communication skills, it is just as important to take into account the need to understand stakeholders and their feedback. Haug described that part of big data science quality relies on the quality levels of data sets [18], and that stakeholders play a role in the evaluation and quality of big data science [19]. Collaboration is fundamental to the success of agile. However, the mechanism of feedback is not often described in much detail [20]. Larson stressed that emphasizing expectations and increasing communication with stakeholders reduces many project problems [21]. Involving stakeholders internally and externally of the team promotes the spirit of agile development, an open-ended process using group communication and fostering an attitude of trying again and through multiple iterations [22]. Thus, stakeholders and end users, are critical for providing and communicating quality feedback to big data projects, especially user expectations.

A recent study in 2017 adapted a communication theory, media richness theory, in agile development using prototypes [23]. The study transferred agile development usually done in the software industry to physical product development. The researchers found that the iterations involving valuable feedback should be reinterpreted to understand validated learnings about the product requirements and realize technical possibilities to a specific design. Although this study examined agile development of physical products, it presents an example of how communication theory can be adapted and paired to agile development. Similarly, our study emphasized the importance and complexity of communication in software development. Additionally, our study is theoretically contextualized in the motivation to further diffuse CI tools as innovations in e-science.

Based on the literature review above, it is evident that feedback is a critical mechanism in big data projects, especially one that employs agile software development methodology. Feedback is often discussed as important, but what makes a feedback message good and productive is usually not clearly defined. Therefore, we pose the research questions (RQs): RQ1, "Why is feedback important for big data projects to build cyberinfrastructure in e-science?" and RQ2, "What are the characteristics of good feedback for big data projects to build cyberinfrastructure in e-science?"

III. Methods of Data Collection & Data Analysis

We conducted 120 semi-structured interviews with 108 informants. One of the interviews was conducted with more than one informant in the same interview, and six informants were interviewed more than once for follow-ups. Therefore, there is a difference between the total number of informants and the total number of interviews conducted.

The interviews were conducted between 16 November 2013 and 25 July 2018. Gender representation is three times more men (*n*= 82) than women (*n*= 26), partly because there are more men than women in the e-science community. Furthermore, their professional roles are represented by a diverse range of stakeholders, including scientists as users (*n*=21), technologists as developers (*n*=38), center administrators as facilitators (*n*=25), liaisons as outreach educators (*n*=6), and also informants who wear multiple hats (*n*=28). Most informants had earned a graduate degree, and many had received a Ph.D.

Through online searches, we created a sampling frame using lists of conference presenters at several conferences related to big data projects in e-science and CI, such as Supercomputing (SC) and Extreme Science & Engineering Discovery Environment (XSEDE, later evolved into PEARC, Practice & Experience in Advanced Research Computing). Our research team conducted 16 in-person interviews at SC Denver 2013, 17 interviews at SC New Orleans 2014, 33 interviews at XSEDE Atlanta 2014, and 10 interviews at PEARC 2018. In total, we conducted 76 in-person interviews at conferences, and 44 telephone interviews throughout the rest of the active project years. The interviews averaged about 37 minutes in length.

Recruitment relied on both random and snowball sampling techniques. First, we randomly selected a row from the spreadsheet containing the complete list of conference presenters in our sampling frame. Second, at the end of each interview, we asked the informant to recommend colleagues we could contact to snowball recruit. In order to increase diversity, we asked specifically, "*Based on this interview, are there any other peers that you can recommend to us who might be interested in participating in this study? We are especially interested in people who are professionally and/or demographically different from you*." For all the pre-scheduled interviews, we briefly researched the informant's professional background online, to identify their professional title and the scope of professional work. Then an email was sent with an invitation to participate in an interview at the conference or over the telephone.

The focus on feedback in this paper is a portion of a larger study on big data projects, CI, and e-science. Therefore, we only describe the interview questions related to the present paper. Our interview questions for feedback management were as follows. The interviewer read a short prompt that defined feedback as "a communication mechanism during software or computational tool development for CI". Informants were then asked, *"What is the role of feedback in software development in cyberinfrastructure and e-science projects?*" and, *"How do you discern between bad and good feedback?*"

The present study took a qualitative social science approach, in that data collection and analysis followed the principles and strategies of grounded theory [24; 25]. The purpose of a grounded analysis is to investigate a phenomenon in its early stages, before a quantitative analysis is feasible. As a research strategy for exploratory research, grounded theory analysis may appear anecdotal on the surface. Therefore, in the next paragraph, we explain how a grounded theory analysis was performed systematically in our paper.

The grounded theory approach includes the use of bottom-up coding methods to identify themes and concepts. Interviews were transcribed using a contract transcription service. The resulting verbatim transcripts were loaded into the qualitative analysis program NVivo [26]. In the present study, we began by *selective coding*, which involved flagging a large portion of interview text related to feedback, communication, and feedback management in the transcripts. Then the analysis continued with *open coding*, which allowed concepts and themes to organically emerge during the careful interpretation process. Through constant comparisons, concepts and themes were either combined or further differentiated, until clear categories were determined. Finally, we engaged in *axial coding*, which related the concepts and themes through specific relationships, such as what resulted in the subsets as answers to the two stated RQs. We report the research findings next.

IV. Research Findings to RQ1: The Importance of Feedback in Big Data Projects in e-Science

We begin this section with reporting on the findings to RQ1, "Why is feedback important for big data projects to build cyberinfrastructure in e-science?" This question is important because strong rationales and logical reasoning can increase project's responsiveness to user feedback, hence improve CI tools, which can further promote their adoption and diffusion in the user community. The general findings suggest that feedback is important because it helps developers in big data projects understand users' needs, make tools user-friendly, prevent emergencies, and receiving any feedback is better for developers than no feedback. We elaborate on these themes below.

A. *Feedback Helps Developers Understand Users' Needs*

Big data projects in e-science often aim to develop CI tools to enable computational research with intensive data in the scientific community. These projects tend to focus on very specific research needs and scientific problems. Ultimately, the projects are funded, developed, and implemented to help a broader community, advance scientific discoveries, and solve large-scale problems. A critical challenge is to understand the needs of users in the community. A technologist from Texas shared,

> [S]ometimes people build stuff because they think it's just a great idea…. [O]ften though, the thing that you set out building isn't the thing that you wind up building… [I]t's very difficult to understand… in a bubble… what's going to meet the needs of a wider range of people

This informant pointed out the danger of being in a 'bubble.' The 'bubble' metaphor refers to developing a CI tool in isolation for an imagined user community, which is far from the actual user community in reality. Instead, a better approach is to actively engage with actual users and seek their feedback early and continuously. A computational technologist from Illinois told us his strategy,

> People start using it, and then they start giving you questions. It's like, 'I would like to do this,' 'Why doesn't it do that?' … And you go back and you look at it, and you say, 'Oh, we never thought anybody would use it like that,' and 'We don't have an algorithm for that yet, but we have an idea'… And we got just incredibly valuable feedback about what was important in the community, where was research needed, which we would not have gotten without people using the stuff we had done and giving us that kind of feedback.

Feedback is essential for helping developers identify and understand what users need to do their research. This understanding is often difficult to identify without involving users from the start. If developers build their tools void of users' feedback, they end up building tools that fail to meet the needs of the users, impeding adoption and diffusion.

B. Feedback Helps Make Tools User-Friendly

If a tool meets the needs of the general community, another factor that impacts the success of the tool is its usability, or user-friendliness. Most users will not take the time to learn in the short-run, and continue using a difficult tool in the long-run, if it lacks usability. Making a tool user-friendly is as important as making sure that a tool meets the needs of the users. In fact, a technologist from the UK commented,

> You can create the best piece of software in the world that mathematically solves the problem that they [the scientists who are users] have. But if it's not usable, if it's not accessible, then you know, you might as well just lock it in a drawer and forget about it because they're not going to touch it.

This informant strongly emphasized the need to attend to usability of the tool. Such an understanding can be accomplished through user feedback and usability tests. However, developers need to be proactive in seeking such feedback, early and often.

C. Feedback Helps Prevents Emergencies

Early feedback is also important for identifying a problem when it is still small, before it becomes an emergency. A problem identified early can save developers time, efforts, and resources to address it. However, in order to identify a problem when it is still small, it takes open and proactive communication to seek early feedback. A center administrator and liaison from Illinois captured this point by sharing her approach,

> Finding ways to open communication so that you are getting feedback and also being proactive about getting feedback. Meeting with stakeholders, important end-users, and regular end-users before things happen, before the complains come, just so you can see if there are weird symptoms that are out there.

The informant highlighted that facilitating open communication involves the efforts by project teams to meet with users as a preventative measure. When talking to users proactively, the developers can learn about potential issues when they are still in early stages.

D. Any Feedback is Better than No Feedback

While receiving feedback is not always a pleasant experience, especially when the feedback is a complaint about what does not work for the users, many informants believe that receiving any feedback is better than receiving no feedback. Without any feedback, developers in big data projects do not know what users are experiencing, and the issues they are facing. A technologist and liaison from Utah told us,

> The worst thing you can do is when you have no feedback… When you have no feedback, you don't know what is happening… You want to push to get feedback—positive or negative—… You can say this is good and [you] can remain in the status quo, or here are the next steps in becoming better.

Many informants acknowledged that receiving no feedback can really be a problem. However, they also commented on users' tendency to not offer feedback. A scientist and an administrator from Indiana talked about this tendency of no feedback from users well,

> We don't hear back from anybody once they're doing research unless there's a problem, and that's a lot of [the] times. It can either mean things are going really well or that they're having problems and [they are] just not telling [you].

A technologist from Michigan further elaborated on the reason behind why sometimes developers do not receive any feedback. He explained that sometimes users assumed that the developers already knew about the problem. He recalled,

> I can't tell you how many times a user said, 'I found this problem, but I thought you knew about it. And so I didn't tell you.' And that is extremely frustrating. I would rather hear about [the same problem] 100 times than not hear about it.

It is important for developers to keep in mind that while many users are not always active in volunteering feedback, developers can be on the lookout for the friendly users in the minority. A computational biologist from Louisiana explained his motivation of being a friendly user, and being involved in the feedback process,

> I am the user, and oftentimes I [say] to the computer scientist, 'I know this is possible. I want to do it, [but] it shouldn't be my problem. You are the developers. You make it happen. I want to see [my research] results'… I'm interested enough in being able to do this routinely that I think it is worth my time to participate in the [software] develop[ment] process, or else it will just never happen.

The findings for RQ1 suggest that feedback is critical to the success of big data projects. However, often times users do not provide feedback at all. When they do, they may not provide feedback in a 'good' way, one that is productive towards solving the problem. This observation led to the second RQ we explored in the next section.

V. Research Findings to RQ2: Characteristics of Good Feedback

RQ2, "What are the characteristics of good feedback for big data projects to build cyberinfrastructure in e-science?" Good feedback makes resolving the issues more efficiently and effectively. This question is important because good feedback can help improve analytics technologies, as without effective CI and computational tools, the accuracy and meaningfulness of analytics results are compromised. By 'good feedback,' informants generally mean when feedback is useful and easy to receive. Overall, six themes emerged to define good feedback as structured, specific, actionable, timely, generalizable, and delivered in an interpersonally tactful way.

A. Good Feedback is Structured

The first characteristic is being structured. Informants suggested different feedback structures, but the resounding theme is for feedback to be structured. A structured feedback message organizes information is some sequences. For example, a feedback message can be organized to explain a user's perspective, articulate his/her assumptions underlying the actions, describe the problem encountered, and explain why the outcome is a problem. Another feedback sequence can start with a description of what happened, followed by when it happened and who was affected. Good feedback presents information in a way that conveys what the users know to the developers who were not present when the problem occurred.

However, a good feedback message should also be structured with high signal-to-noise ratio, as a scientist-developer and co-producing user in California suggested, avoiding the "TL; DR Effect". TL; DR means 'Too long; Didn't read'. In other words, a good feedback message is designed to convey the user's standpoint with the appropriate amount of information, so developers are not overwhelmed by the feedback received.

B. Good Feedback is Specific

The second characteristic is being specific. Specific feedback describes why something did not work and is illustrated by a test case. It includes examples of how the software has and/or has not worked. A technologist and liaison from Utah further explained that a good feedback message also describes the project and the motivation behind it, and what the user is trying to accomplish with the software for the project. He stated, "You have the *what* but describe to me the *why*?" That way, developers will have enough information to not only fix the reported error, but to help the users become successful.

Good feedback should be specific, because with specificity, the developers can reproduce the error. A technologist and a co-producing user from Louisiana went as far as stating, "I don't bother to report something if I can't tell people how to reproduce it." Users with problems often forget to provide enough information to get the help they need. Often the problem needs more information in order to get an answer.

Classic bad feedback that is not specific simply states, 'It doesn't work!' The same informant from Louisiana continued, "You try to get information out of them… Why doesn't it work? What do you mean by, 'It doesn't work'?" He offered an analogy to describe the irony of vague feedback in a joking manner, "An airline problem ticket simply stated, 'Something is broken in the cockpit.' An engineer looked at it and wrote, 'Something is fixed in the cockpit'." If feedback is not specific, it becomes something developers in big data projects cannot act on to improve for users.

C. Good Feedback is Actionable

The third characteristic is actionable. Good feedback should have a clear path to actions towards resolving an error. Good feedback enables developers to take steps towards correcting the error. An administrator and a computational health researcher in Texas explained her belief that a feedback giver should always put him/herself in the shoes of the feedback receiver by reflecting on,

> Am I asking or saying something that someone can then take and do something [about]?... By actionable I don't mean, 'Hey, I think you should do X'... [Instead, it should be] something that you can do something with. But for example, 'My code died!' And that's it. What can I do with that? I know there's problem but there's not direction to go.

Along with the idea of actionable feedback is the notion of feasibility. If a piece of feedback suggests actions that are not feasible for the developers, perhaps due to constraints of time, expertise, and/or resources, the feedback is not helpful. In other words, users would be more effective if they consider feasibility in their feedback to developers about a problem.

D. Good Feedback is Timely

The fourth characteristic is timely. In general, developers prefer that users report an error as soon as it happens. An informant explained that developers know that most feedback is spontaneous and they are prepared to deal with it as it arises. Another informant, who is an administrator and liaison from South Dakota further elaborated,

> It's just kind of the nature of working with researchers [who] are so busy. And a lot of times grad[uate] students are given this work to do, and they're not sure which application to run. I'll find out [the technical problem] when it's too late.

Delays as described above are not ideal and productive for developers. According to the informants, delayed communication is usually the tendency of faculty principal investigators (PIs). For graduate students, an administrator and liaison from Illinois discussed her observation,

> So grad[uate] students are really good about giving us feedback quickly because it affects their day to day work. Faculty might say, 'My grad[udate] student had an issue with this a month ago' and have no other details. And [giving details] may be hard if they [faculty] are not in the day to day research.

What the informants reported was that feedback timeliness and the sense of urgency depends on how the problem affects the users. While faculty PIs are in the position to offer feedback to developers, faculty are actually removed from the nitty gritty of the research work, thus not having as much detail to offer about the feedback, and not feeling the time pressure to receive help as in the case of their graduate students.

E. Good Feedback is Generalizable

Good feedback is generalizable and has broader impacts. Sometimes feedback from users can be narrow and/or simply a personal preference. For example, an administrator and a computational health researcher from Texas clarified,

> Because sometimes you'll get people say 'I don't like blue' or 'Why is this button like this?' That's not particularly actionable. What I would consider good feedback be someone who saw something perhaps isn't [a] personal preference, but thought this also extends to the broader community.

Good feedback should be generally applicable to the broader community, not simply the individuals who take the time to contact the developers. This characteristic is important to note because there could be countless personal requests to the developers, and it is practically impossible to include every personal request. Given the inherent limitation of time and resources on the developers' end, they have to make tough choices to exclude narrow feedback in favor of generalizable ones that can better benefit the broader user community.

F. Good Feedback is Delivered in a Tactful Way

Finally, good feedback is delivered in an interpersonally tactful manner. In general, developers are humans with feelings and emotions. They tend to respond well to feedback that is framed in a palatable way. For example, an administrator and a technologist from Utah suggested an approach using an analogy,

> Like in giving medicine, there's an old adage that 'You start with something sweet.' Focus on the positive pieces. Make sure to question your own understanding when you're giving tough feedback to the person. Say, "Hey this is my perception of your service or your work, am I correct?" And if they tell you something and you don't understand. Mirror it back to them and try to understand what you heard is what they really meant.

A scientist-developer and co-producing user from California advised that in general, good feedback would avoid being emotionally driven, full of rage, simply venting, and/or turning the feedback into a personal attack. Good feedback goes beyond reporting the problem and shuns conveying a demanding expectation and/or a negative attitude to the developers in big data projects.

VI. Conclusion, Limitations, and Implications

In this paper, we set out to pursue the two research foci of why feedback is important for big data projects in e-science, and what makes a piece of feedback regarded as 'good feedback' in these projects. With a grounded analysis of interview data, we concluded that feedback helps developers understand users' needs, makes tools user-friendly, prevents emergencies, and is better for developers than no feedback. Furthermore, good feedback is often structured, specific, actionable, timely, generalizable, and delivered in a tactful way.

A. Limitations

While we believe the findings in this paper contributes to the research conversation on methodologies of big data projects, especially in developing computational tools and CI in e-science, we note three particular limitations of our study. First, interviewing solicited descriptive data, based on the rationales and retrospective sense-making of the informants. Our findings are not meant to be causal claims, but exploratory in nature. Therefore, we note the limitation that feedback with all six characteristics may not be well-received by all developers.

Second, this paper does not include an experimental evaluation. To do so would require a different type of data, collected in a lab setting. However, such a future effort would be worthwhile, especially in addressing the first limitation of not able to make causal claims, based on the findings. An example of such an experiment may include a comparison between an experimental group with feedback training vs. a control group. Another example would be to measure a CI tool's adoption/diffusion potential, as accounted for by a project's responsiveness to user feedback.

Third, CI's complete vision was for computational tools and e-science technologies to enable both research and education in science and engineering [1]. However, due to the early phase of CI development during data collection, and the conferences from which we recruited informants, the findings are mostly about big data projects for scientific research, instead of both research and education.

B. Practical & Theoretical Implications

What are the implications of this study? We suggest several below. Practically and first, feedback improves the quality of big data technologies developed, in terms of meeting users' needs and having usability. Therefore, developers would be wise to proactively solicit and gather feedback from users, early and often. Being proactive is important because many users have the tendency to not provide feedback. In such a case, receiving no feedback is really the worst situation for the developers. This rationale, along with how feedback can help improve needs assessment, usability, and emergency prevention, should be taken seriously by developers in big data projects. Developers can proactively seek out feedback by identifying friendly users and frequently engage with them. Such an approach would greatly increase user feedback in terms of quantity.

Second, given that constant user feedback helps agile software development be optimal, developers can use the six characteristics of good feedback as prompts in collecting input from users both online and offline. For example, when using an online bug tracker or GitHub to submit errors, users can be prompted, and their submission can be guided, by the six characteristics. Similarly, the six characteristics can be used by developers to guide their phone calls and/or face-to-face meetings with users, especially when the users are internal or in-house. This is especially helpful for users who have really never been trained in providing good feedback, which is likely to be for most, if not all, users. Such an approach would greatly increase user feedback in terms of quality.

Third, given the importance of good feedback and how users may not be interpersonally tactful in providing feedback, it makes sense to include feedback management in training for software engineering project managers. Similarly, such a training can be taught in college classes as part of the project management curriculum, software engineering curriculum, and/or workshops at conferences where developers in big data projects are likely to attend for professional development. We see this paper as the first step towards promoting more formal training to improve feedback communication in software engineering and big data projects. Such an approach would greatly increase user feedback in terms of process management.

Theoretically, the fourth implication suggests that research on agile software development needs to move beyond simply noting that feedback is important but delving deeper into the kind of feedback that maximizes the potential of the agile methodology. Bad feedback, such as feedback that is unstructured, vague, non-actionable, delayed, narrow, and delivered as a personal attack in an emotional driven way, will do more damage than good to a big data project. While our findings for RQ1 did suggest that any feedback is better than no feedback, we believe that is true mostly with developers who are highly skilled in feedback management, and who have the ability to process bad feedback in a constructive way. Therefore, good feedback, as defined by the six characteristics identified in this paper, optimizes the agile process, and should be theorized as the conditions for effective agile software development.

Fifth, feedback is in fact key to diffusion of dynamic innovations. In diffusion of innovations theory [27; 28], researchers often investigate static innovations, which were pre-designed, mass-produced, bought off-the-shelf, and use-as-instructed after full product development. As can be gleaned from this paper, CI and computational tools developed out of big data projects in e-science are often user-driven, custom-made, produced on demand, permanently beta, constantly evolving as the tools are being adopted and implemented. In fact, such a dynamic nature is key to the tools' ability to survive and sustain their diffusion. When a tool stops evolving against the backdrop of the constant shifting needs of the user community, the tool will die by the way side. The key mechanism to keep a tool evolving appropriately is feedback. We advanced the argument that feedback is a key theoretical mechanism to understand today's dynamic innovation diffusion.

In summary, feedback is critical to big data projects, and developers can implement the six characteristics of good feedback as prompts when seeking feedback from users, especially when the goal of the projects is to develop analytics technologies that can diffuse and be adopted within the broader user community. The implications for feedback illustrate that feedback is a communication skill that can be developed not only practically for projects but also for the diffusion of the technologies. Theoretically, we brought together three strands of research literature of agile software development, diffusion of innovations, and organizational communication through the research focus of feedback in big data projects. Further research should explore the common communication channels for feedback, strategies for feedback management, and what conditions support the effectiveness of these feedback channels and management strategies, in order to expand on how user feedback can be further promoted and harnessed to optimize successful outcomes of big data projects.

ACKNOWLEDGMENT

We acknowledge the early contribution of Mona Sleiman, Brett Robertson, Christian Keroles, Michelle Williams, Dominique Stewart, Andrew Schrock, and all the research team members who participated in the data collection for this project. We also thank Rion Dooley, Nancy Wilkins-Diehr, and John Towns for their support of this project.

REFERENCES

[1] D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. Garcia-Molina, M. L. Klein, D. G. Messerschmitt, et al., "Revolutionizing Science and Engineering through Cyberinfrastructure: Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure",

National Science Foundation, 2003, http://www.communitytechnology.org/nsf_ci_report/.

[2] K. F. Kee and L. D. Browning, "The dialectical tensions in the funding infrastructure of cyberinfrastructure," Computer Supported Cooperative Work, vol. 19, August 2010, pp. 283-308.

[3] C. E. Shannon and W. Weaver, The Mathematical Theory of Communication, Urbana, IL: University of Illinois Press, 1949.

[4] J. Fiske, Introduction to Communication Studies, 2nd ed., London: Routledge, 1990, pp. 6-22.

[5] J. Fiske and H. Jenkins, Introduction to Communication Studies, New York: Routledge, 2011.

[6] J. Waldeck, "Feedback," International Encyclopedia of Organizational Communication, C. R. Scott and L. K. Lewis, Eds., Hoboken, NJ: John Wiley & Sons, 2017, pp. 899-919.

[7] D. R. Ilgen, C. D. Fisher, and M. S. Taylor, "Consequences of individual feedback on behavior in organizations," J. of Applied Psychology, vol. 64, 1979, pp. 349–371.

[8] M. S. Taylor, C. D. Fisher, and D. R. Ilgen, "Individual's reactions to performance feedback in organizations: Control theory perspective," in Research In Personnel And Human Resource Management, K. M. Rowland and G. R. Ferries, Eds. Greenwich, CT: JAI Press. 1984, pp. 81–124.

[9] M. London and J. W. Smither, "Feedback orientation, feedback culture, and the longitudinal performance management process," Human Resource Management Review, vol. 12, 2002, pp. 81-100.

[10] S. E. Moss, E. R. Valenzi, and W. Taggart, "Are you hiding form your boss? The development of a taxonomy and instrument to assess the feedback management behaviors of good and bad performers," J. of Management, vol. 29, 2003, pp. 487-510.

[11] S. J. Ashford and L. L. Cummings, "Proactive feedback seeking: The instrumental use of the information environment," J. of Occupational Psychology, vol. 58, 1985, pp. 67-79.

[12] J. van der Rijt, P. Van den Bossche, and M. S. R. Segers, "Understanding informal feedback seeking in the workplace: The impoact of position in the organizational hierarchy," European J. of Training and Development, vol. 37, 2012, pp. 72-85.

[13] J. B. Walther, Y. J. Liang, D. C. Deandrea, S. T. Tong, E. L. Spottswood, and Y. Amichai-Hamburger, "The effect of feedback on identity shift in computer-mediated communication," Media Psychology, vol. 14, 2011, pp. 1-26.

[14] D. K. Becker, "Predicting outcomes for big data projects: Big data project dyanmics (BDPD)," Proceedings of the 2017 IEEE International Conference on Big Data, pp. 2320-2330, December 2017.

[15] J. S. Saltz, S. Yilmazel, and O. Yilmazel, "Not all software engineers can become good data engineers," Proceedings of the 2016 IEEE International Big Data, pp. 2896-2901, December 2016.

[16] L. Williams, and A. Cockburn, "Agile software development: It's about feedback and change," Computer, vol. 36, 2003, pp. 39-43.

[17] M. Lycett, R. D. Macredie, and R. J. Paul, "Migrating agile methods to standardized devleopment practice," Computer, vol. 36, 2003, pp. 79-85.

[18] J. S. Saltz and I. Shamshurin, "Does pair programming work in a data science context? An initail case study," Proceedings of the 2017 IEEE Internationa Conference on Big Data, pp. 2348-2354, December 2017.

[19] F. S. Haug, "Big bad data science," Proceeding of the 2016 IEEE International Conference on Big Data, pp. 2863-2871, December 2016.

[20] D. Larson, "BI principles for agile development: Keeping focused," Business Intelligence J., vol. 14, 2009, pp. 36–41.

[21] D. Larson and V. Chang, "A review and future direction of agile, business intelligence, and data science," International J. of Information Management, vol. 36, 2016, pp. 700-710.

[22] J. Saltz, "The need for new process, methodologies, and tools to support big data teams and improve big data project effectiveness," Proceeding of the 2015 IEEE International Conference on Big Data, pp. 2066-2071, October 2015.

[23] T. S. Schmidt, A. I. Böhmer, A. Wallisch, K. Paetzold, and U. Lindemann, "Media richness theory in agile development choosing appropriate kinds of protoypes to obtain reliable feedback, 23th International ICE Conference on Enginnering, Technology and Innovation (ICE'17), pp. 521-530, June 2017.

[24] J. Corbin, and A. Strauss, "Grounded theory research: Procedures, canon, and evaluative criteria," Qualitative Sociology, vol. 13, 1990, pp. 3-21.

[25] B. G. Glaser and A. Strauss, The discovery of grounded theory: Strategies for qualitative research. Piscataway, NJ: Aldine Transaction, 1967.

[26] P. Bazeley and K. Jackson, Qualitative Data Analysis with NVivo, Thousand Oaks, CA: Sage, 2011.

[27] E. M. Rogers, Diffusion of Innovations, 5th ed., New York: Free Press, 2003.

[28] K. F. Kee , "The ten adoption drivers of open source software that enables e-research in data factories for open innovations," In Big data factories, computational sciences: Collaborative approaches, S. A. Matei, N. Jullien, and S. P. Goggins, Eds., New York: Springer, 2017, pp. 80-102