The Appropriation of GitHub for Curation

Yu Wu[1], Na Wang[2], Jessica Kropczynski[3], and John M. Carroll[4]

[1,3,4]Information Sciences and Technology, Penn State University, University Park, PA, United States
[2]Samsung Research America, Mountain View, CA, United States

Corresponding author:
Yu Wu[1]

Email address: yuw132@psu.edu

ABSTRACT

Github is a key online collaborative software development environment. In this paper we describe a new category of Github project: Curation projects collect, evaluate, and preserve resources for software developers. We investigated 1) what motivates software developers to engaged in curation; 2) how software developers benefit from curation activities; and 3) how well GitHub supports/fails to support curation practices. We conducted in-depth interviews with 16 software developers each of whom host curation projects on GitHub. Our results suggest that software developers' motivations for curation on GitHub are similar to their participation in open source projects. Convenient tools (e.g. Markdown syntax and Git version control system) and the opportunity to address professional needs of interests of large numbers of peers attract developers to engage in curation projects. Software developers benefit from curation projects through learning opportunities, support for development work, and professional interaction. However, curation is limited by GitHub's document structure & format and also its lack of key features, such as search. We discuss design possibilities to encourage and improve curation appropriations of GitHub.

INTRODUCTION

GitHub is a collaborative coding environment that employs social media features. It encourages software developers to perform collaborative software development by offering distributed version control and source code management services with social features (i.e., user profiles, comments, and broadcasting activity traces) (Dabbish et al., 2012). This web-based tool has attracted significant attention from both industrial and academic communities. By the end of 2012, software developers hosted over 4.5 million repositories on GitHub (Marlow et al., 2013). It has not only topped the list of preferred software hosting and collaboration services among developers (Doll, 2013), but also inspired a number of researchers to investigate how its features have supported software development practices (Dabbish et al., 2012; Marlow et al., 2013; Singer et al., 2013). Specifically, prior studies have uncovered how software developers make social inferences and collaborate with each other over GitHub social features (i.e. activity traces and follow function) (Dabbish et al., 2012; Marlow et al., 2013; Wu et al., 2014).

In addition to hosting and collaborating in GitHub repositories, a new category of practices has recently emerged — software developers have begun appropriating GitHub repositories to create resources lists and make them public (Wu et al., 2015). Such practices are recognized as curation—activities to select, evaluate, and organize resources for preservation and future use (Duh et al., 2012). In 2014 and 2015, curation repositories on GitHub gained enormous popularity. The number of curation repositories increased, and many of them are among the most famous repositories on the entire platform (Wu et al., 2014). In light of the broad popularity of curation practices on GitHub, one might expect that they are well-understood. However, research in this area is relatively sparse. The investigation of curation practices on social media has only begun recently, and it remains under-explored in general (Duh et al., 2012). The existing curation literature focuses on microblogging services (i.e., Twitter) (Duh et al., 2012; Dabbish et al., 2012; Greene et al., 2011) and media sharing service (i.e., Pinterest), leaving untouched as an area of exploration the nature of curation in software development practices.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2952v1 | CC BY 4.0 Open Access | rec: 28 Apr 2017, publ: 28 Apr 2017

To address this gap in the literature, we conducted semi-structured interviews with 16 participants to probe the curation practices on GitHub. To be more precise, this study aims to investigate: 1) what are developers' motivations that drive curation practices, and why GitHub is chosen for this purpose? 2) Why and how are curated resources used? 3) And what are current limitations and potential future improvements for curation on GitHub? Our results suggest that curation practices on GitHub mostly grow out of software developers' internal (altruism) and extrinsic motivations (personal needs and peer recognition). Software developers choose GitHub to perform curation practices mainly because this platform provides convenient tools and attracts vast groups of people with common interests. Software developers also benefit from curation in many aspects, including but not limited to better software development support, and more efficient learning and communication approach. Further, curation represents a case that a collaborative working space is appropriated to an end-product for communicating high quality resources, suggesting GitHub repositories can be used for communication purposes to support software developers' community. However, current curation practices are restricted by document format and curation process, and are bounded by GitHub features as well. More built-in tools, such as navigation support within curation projects and automated resources checking and evaluation, holds potential for improving current practices. Our study contributes to better understanding of software developers' motivations to curate resources and the nature of appropriating GitHub for curation purpose. We also discuss design implications that may better support this practice in the software developers' community.

BACKGROUND

This section reviews the past literature on motivations to curate, tools for curation, and current curation practice on GitHub.

Motivations of Curation in Social Media Era

Curation is a common practice in Archaeology. It is the activity of collecting, evaluating, organizing, and preserving a set of resources for future use (Bamforth, 1986). In the Internet era, curation is commonly referred to by librarians and archivists as "digital curation" to preserve digital materials (Higgins, 2011). It shares characteristics of social bookmarking behavior, where users specify keywords or tags for Internet resources, and organize and share these resources with others (Farooq et al., 2007). Early popular social bookmarking tools include del.icio.us, which allows sharing of personal bookmarks (Golder and Huberman, 2006); Flickr, a photo tagging and sharing service (Marlow et al., 2006); and Reddit, a community-driven link sharing, comment, and rating service (Singer et al., 2014). Curation behaviors have been further studied since social media has enabled new forms of curation. Specifically, Duh et al. (2012) report the use of a third party tool, Togetter, for curating tweets, and uncovering the intended purposes for these curated lists, including recording a conversation, writing a long article, or summarizing an event (Duh et al., 2012). Zhong et al. (2013) has conducted surveys of Pinterest and Last.fm users and found that the majority users engage with the curation site for personal interests rather than for social reasons (Zhong et al., 2013). A recent study examined the ways that communities leverage a variety of social tools for curation to support vital community activities in a large enterprise environment (Matthews et al., 2014). The authors also call for future studies on curation in public Internet communities (Matthews et al., 2014).

Curation on GitHub is different from the above studies in the following sense. First, the user body of GitHub is drastically different. Services like Twitter, Pinterest, and Reddit, are services for the general population with diverse backgrounds and interests, while GitHub is intended for a focused community of software developers. Members of the software developers' community share a set of common goals and practices, which is likely to affect their participation in curation practices as well. Second, unlike Pinterest, which itself is designed for curation purpose, GitHub is an online work platform designed for software developers to collaborate with others on software projects, and curation is an appropriation of the collaborative coding features of the platform. The reasons behind such appropriation and whether GitHub features meet curation needs of developers are yet to be discovered. Third, the technologies affordances of GitHub drastically depart from the above mentioned services. Tools like Pinterest and Flickr, are for personal collection and sharing. Reddit allows users to vote to promote links, but it hardly preserves resources. GitHub provides a collaborative working space, i.e. repository, where software developers can work on the same project together and are enforced by Git workflow. Therefore, GitHub is distinct regarding user base, intended purpose, as well as technology affordances. Its appropriation for curation purpose raises an interesting question concerning user's motivations and experiences.

Software Developers' Motivations in Participating Online Communities

Researchers report two main categories of motivations that drive software developers' voluntary participation in open source projects: 1) internal motivations, i.e. intrinsic motivations, altruism, and community identification, and 2) external rewards, including expected future rewards and personal needs (Hars and Ou, 2001; Ye and Kishida, 2003). Internal factors include "intrinsic motivation" refers to software developers motivation by the feeling of competence, satisfaction, and fulfillment in participating in open source; "altruism" refers to software developers desire to care for others' welfare at own cost; and "community identification" refers to software developers' alignment of goals with the larger community. External factors include "future rewards" when software developers view their participation as investment, and expected future returns, including revenues from related products and services, human capital, self-marketing, and peer recognition; "Personal needs" are software developers' personal demand for their activity, for e.g., Perl programming language and Apache web server both grew out of software developers' self-interests to support their work (Hars and Ou, 2001). Both internal and external factors are important motivations that drive software developers' participation in open source projects.

The rise of social media affects the way software developers participate in online space. "Social media" in software developers' community is often referred to as "socially enabled" tools, where social features are added to software engineering tools (Storey et al., 2014). It lowers the barrier to publishing, allows fast spreading, and enables communicating at scales, which facilitate a "Participatory Culture" in the software developers' community (Storey et al., 2014; Jenkins et al., 2009). As a result, software developers increasingly participate in the community with social media for learning, communication, and collaboration (Dabbish et al., 2012; Doll, 2013; Singer et al., 2013). However, software developers' motivations for participation remain similar: they are motivated to participate for personal needs (e.g., improve technical skills) and peer recognition (e.g., get recognized by the community) (Storey et al., 2014).

Despite the well-studied motivations for software developers' participation in the online community, software developers' motivations in curation practices with an appropriation of a collaboration software development tool are currently under-explored.

Prior GitHub Research

In recent years, GitHub has drawn enormous attention from researchers. It features transparency, such as activity traces, user profiles, issue trackers, etc., in source code hosting and collaboration (Storey et al., 2014; Dabbish et al., 2013). Researchers examined in details about how such transparency allows software developers to engage with software practices in the community (Dabbish et al., 2012; Doll, 2013; Singer et al., 2013). For example, Dabbish et al. (2012) found that the activity logs and user profiles on GitHub motivate community members to contribute to software projects (Dabbish et al., 2012). Marlow et al. (2013) discovered that developers use a variety of social cues available on GitHub to form impressions of others, which in turn moderate their collaboration (Marlow et al., 2013). Singer et al. (2013) put GitHub in a larger social media environment, and learned that software developers leverage transparency of socially enabled tools across many social media services for mutual assessment (Singer et al., 2013).

These studies focus on how the technology affordances of GitHub and other social media affect software practices, including learning, communication, and collaboration (Storey et al., 2014). Curation as an emerging practice on GitHub raises interesting questions concerning the reasons such practice thrive in the software developers' community, and more specifically, why it emerged on GitHub and whether GitHub features fully support this type of practice.

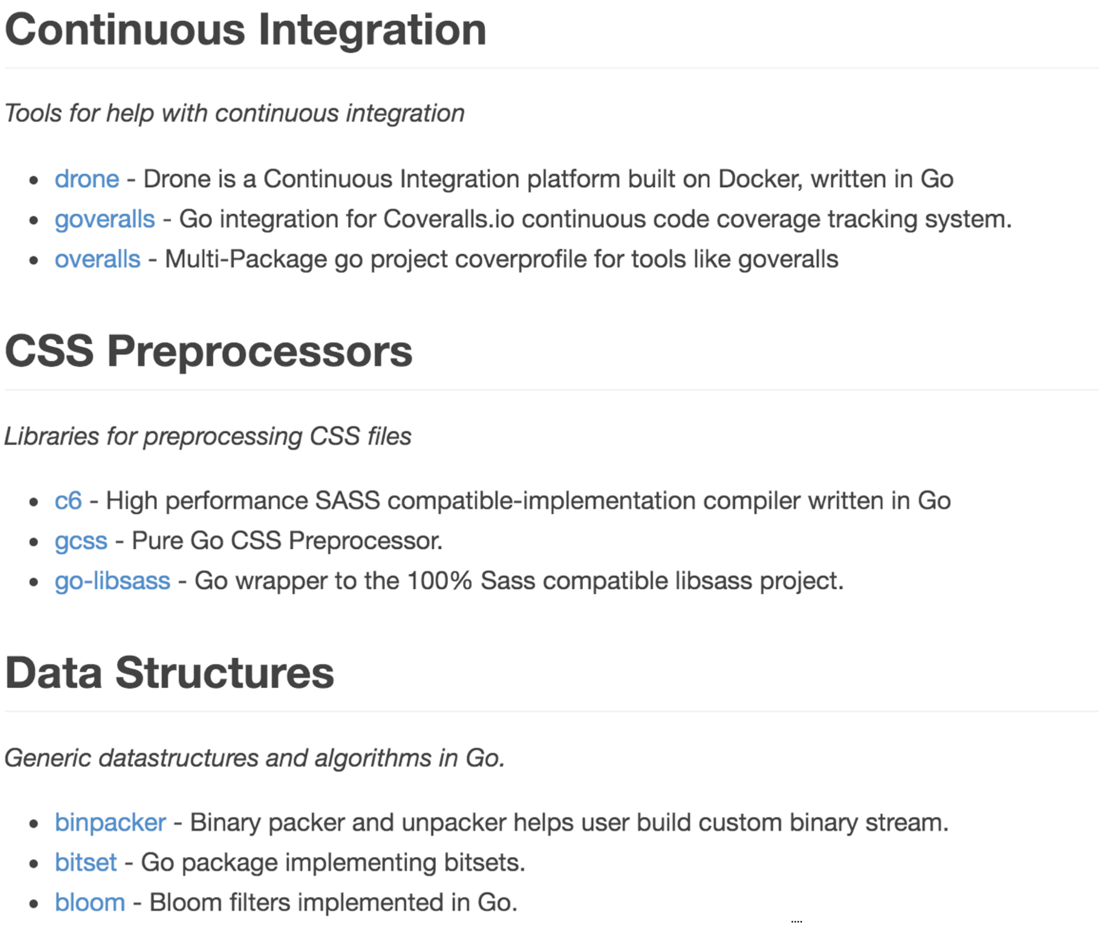
Appropriating GitHub for Curation

Curation practices are enabled by GitHub features. Specifically, GitHub introduced a README.md file in the root directory for each repository. The contents of the README.md file are displayed on the front page of the repository, i.e. if one visits the URL of a software repository hosted on GitHub in a browser, the README.md file will be displayed as a web page (Fig. 1) [1] along with repository structure and some project statistics, such as the number of forks and stars (McDonald and Goggins, 2013). The content of

[1] https://github.com/avelino/awesome-go

REAME.md file can be structured with Markdown syntax[2], which provides rich text features, including table of contents, links, tables, etc. README.md is designed for adding description and documentation for a repository [3].

Curation on GitHub appropriates the README.md file of a repository to create a list of resource indexes within one page. It categorizes resources into different themes and differentiates them into sections. Typically, each resource is recorded with the resource name and a brief description of what the resource is (Fig. 1). In addition, URLs are attached to each curated items. Clicking a resource name will direct the user to the real web location of the resource.



Continuous Integration

Tools for help with continuous integration

- drone - Drone is a Continuous Integration platform built on Docker, written in Go
- goveralls - Go integration for Coveralls.io continuous code coverage tracking system.
- overalls - Multi-Package go project coverprofile for tools like goveralls

CSS Preprocessors

Libraries for preprocessing CSS files

- c6 - High performance SASS compatible-implementation compiler written in Go
- gcss - Pure Go CSS Preprocessor.
- go-libsass - Go wrapper to the 100% Sass compatible libsass project.

Data Structures

Generic datastructures and algorithms in Go.

- binpacker - Binary packer and unpacker helps user build custom binary stream.
- bitset - Go package implementing bitsets.
- bloom - Bloom filters implemented in Go.

....

Figure 1. A part of README.md file of awesome-go curation project.

Curation on GitHub appropriates README.md to create a list of resource indexes. It categorizes resources into different themes. Typically, a curation repository contains several sections, and each section contain a set of resources that belong to the same theme. Each resource is recorded with the resource name and a brief description (Fig. 1). In addition, URLs are attached to each curated items. Clicking a resource name will direct the user to the real web location of the resource.

METHODOLOGY

To explore and understand software developers' experiences with the appropriation of GitHub for curation purposes, we conducted a qualitative study with 16 curation project owners. The study was approved by Penn State University Institutional Review Board, under the approval number PRAMS00044217. In this section, we describe our recruitment procedure, interview protocol, and data analysis processes.

[2] https://help.github.com/articles/basic-writing-and-formatting-syntax
[3] https://help.github.com/articles/create-a-repo

Participants Recruitment

To reach out to the correct participants, on 12/07/2015, we queried the GitHub search API with keywords "curated list" to search for curation repositories. The query returned 896 curation repositories hosted on GitHub. By going through the list of these repositories, we recorded the owner's ID for each repository. Then, for each owner's ID, we queried GitHub API again to fetch their profiles with email addresses. It returned 405 unique owners with email addresses. Following that, we randomly sent out 172 email invitations to invite curation project owners for a semi-structured online text-based interview. Upon participants' choices, we carried out the interview via Facebook Messenger, Skype, or Google Hangouts. We began our recruitment process in early December 2015 and completed all interviews in late January 2016.

Our 16 participants included 15 males and one female with GitHub experiences ranging from 6 months to 6 years. 14 of the participants are professional software engineers, one is a graduate student, and one is a microbiologist. 11 participants used the descriptive word "awesome" as the prefix to name their curation repository, which follows a typical naming convention. The participants had varying number of followers: 5 have less than 10 followers; 8 have less than 50 but more than 10 followers, and 4 have more than 50 followers. In the following discussion, we refer to individuals by participant number (from P1 to P16).

Interview Protocol

We conducted a text-based online interview with each participant, spending approximately 30 to 60 minutes in a discussion. The interview questions were semi-structured by the four general themes below.

- Motivations to curate resources,
- Reasons for their technology choice (GitHub),
- How curated lists are useful,
- The limitations of current curation practices (on GitHub).

And the questions we asked were open-ended enough that we could pursue new topics raised by the participant.

Participants were interviewed in English. The interview scripts were then analyzed to discover themes related to cultural differences and communication difficulties.

Data Analysis Procedure

We qualitatively analyzed participants' responses to identify key findings. We conducted our analysis iteratively, carrying out four rounds of interviews and subsequent analysis, allowing the first analysis process to guide our second round of interviews, and then the third and the fourth, allowing themes and codes be identified, discussed, and refined (Lacey and Luff, 2001). We performed the qualitative data analysis of the chat scripts with open coding schema (Strauss, 1987). We concluded the study after reaching the point of theoretical saturation, when categories, themes, and explanations repeated from the data (Marshall, 1996). A second researcher independently coded four sample interviews transcripts. Our analysis showed inter-coder agreement between the two researchers (kappa = 0.73).

RESULTS

Our analysis results reveal the curation practices on GitHub from four aspects, including 1) motivations to curate, 2) technology choice, 3) the use of curated resources, and 4) the current limitations of curation practices on GitHub.

Motivations to Curate

Internal factors (i.e. altruism, community identification, and intrinsic motivation) and external rewards (i.e., personal needs and peer recognition) motivate software developers to participant in open source projects (Hars and Ou, 2001). In this study, our participants confirmed altruism (62.5%), personal needs (93.8%), and peer recognition (31.2%) as their motivations in the emerging curation context.

Internal Factors - Altruism

Participants report that they engaged in curation practices because other community members might benefit from the effort. They believed the high quality curated resources could help beginners to get started with programming:

> "I see so many people when they take introductory classes in programming, they come to GitHub to get ready repositories...and that is overwhelming at first...so to get the started and motivated with programming I thought of collecting resources together in (P3's curation project)" – P3

Altruism is widely recognized as an important motivation for software developers' participation in open source projects (Hars and Ou, 2001; Ye and Kishida, 2003). Its arise among curators indicates that helping each other may be a common attribute for software developers to participate in online activities. Thus, when we design systems for facilitating software development related practices, the system should always allow software developers to support each other.

External Rewards

Personal needs and future rewards form software developers' external rewards to participating in open source projects (Ye and Kishida, 2003). Both of them are also driving forces of engagement in curation practices.

Personal needs is the major reason for participation in curation (98.3%). Specifically, software developers find curation repositories improve productivity and enable communication with others. 50% participants who are very familiar with a particular set of resources, before creating curation projects, they still used search engines every time they tried to locate the exact Internet location of the resources. One of the important reasons they curate resources is to avoid such repeated search effort.

> "Before making the repo I had to do research each time I needed a (P12's curation topic). Now that I have a list, I just refer back to it when needed." – P12

> "I simply created my own list of the sites I found to be good. The idea really was to get out there scout for sites once and then be able to come back to a list without worrying about it having sites I found bad." – P9

In addition, a curated repository has a permanent URL, which is convenient to share with others. Our participants find them usually communicate with others about certain resources. With curation project, they only need to point others to the URLs of their curation repositories, which is both convenient for them to share and for others to find. For example, P14 created the curated list so when she can conveniently point the resources contained in her list to others.

Peer Recognition surface as another important motivation for software developers' participation in curation practices on GitHub.

Software developers' community on GitHub adopts a particular way to endorse curation projects. A highly reputable software developer on GitHub, Sindre Sorhus (9.2K+ followers), created an "awesome" (repository name) project on 07/11/2014, which is a meta list of curated lists [4]. It contains a community drafted "awesome manifesto" [5], which depicts guidelines and standards for curation practices, and requires curation repositories to conform if they want to be included in this meta list. The project currently has around 2500 watchers, more than 35000 stars, and approximately 4000 forks, ranking the 2nd most starred repositories that are created after 01/01/2014 [6]. The "awesome" project itself attracts attention from a large number of community members on GitHub.

11 out of 16 participants in this study used "awesome" as a prefix to their curation project name, which tries to conform to a naming convention as well as indicating the quality of the content. 10 of them mentioned that they were inspired by the original "awesome" project, and 4 of them hoped to get their curation repository indexed by it. One participant's curation project is included in "awesome" list, and he felt that it was a great honor (P10). P12 is currently putting effort forward to improve his curated list to conform to the guidelines and standards as defined by the "awesome" project, so that "...with the Awesome endorsement I'm hoping it becomes a collection people trust" (P12). It demonstrates that our participants are putting efforts to align their goals with the larger community, i.e., conforming to the community standard for curating high quality resources, and would like to be recognized by the community.

[4] https://github.com/sindresorhus/awesome

[5] https://github.com/sindresorhus/awesome/issues/207

[6] https://github.com/search?utf8=%E2%9C%93&q=created%3A%3E2014-01-01+stars%3A%3E1&type=Repositories&ref=earchresults

In addition, P14 reported that her involvement in curation efforts helped her obtain her current job, and P10 reported that a company approached him and wanted to collaborate with him on his curated content. But these rewards emerge as side-effects of curation efforts, not one of the guiding motivations for software developers to begin a curation project.

The Technology Choice - Why GitHub

Compared with GitHub, services like Wikipedia should be better for hosting curation projects by providing convenient editing and collaborating features. However, the curation projects were predominantly hosted on GitHub, whose features and information structures are initially designed for source code hosting and project collaborating (Marlow et al., 2013; Storey et al., 2014) but not creating and preserving lists of resources. In this section, we will address why curation happens on GitHub.

Familiarity with GitHub

Software developers' existing knowledge about GitHub and its features (i.e., their strong media literacy (Storey et al., 2014) with GitHub) prompted them to choose this platform to host their curation projects.

In general, software developers are familiar with GitHub's text editing format (i.e., Markdown syntax) and are comfortable using it. P4 and P7 both claimed that "GitHub was a tool that I was familiar with" and "so yes github would be a more natural tool to use". Specifically, software developers are accustomed to write and format text contents with Markdown syntax. For example, P11 expressed that "... I love write in markdown format!", and P5 considered that "Github has a really easy way to write content in rich format (using Markdown) and view it."

> "... as developer, I think github is the best place for developer to collaborate with other to build good resource." – P15

Intimate knowledge about GitHub collaborating features is another factor:

> "Github is a really good platform to collaborate. Anyone could come, fork it, extend it and ask me to 'Merge' it (update my list)." – P5

> ". . . the advantage of using Github is other people can contribute easily." – P4

Relevant Content and Potential Audience

Participants also choose GitHub for curation because: 1) their curated contents are relevant to GitHub context, 2) there are a lot of potential audience on GitHub, and 3) GitHub encourages contributions.

15 out of 16 participants' curation projects are related to software development practices. They consider GitHub just suitable as a platform for software developers to collaborate on curation: "(it is) ... the place to be for projects like this" (P2).

GitHub has attracted a large base of like-minded users when it comes to software development, which increases the chances of matching with an interested audience:

> "GitHub has a very large audience / devs actively spending time in it, so it's definitely the right place to publish a project such as this..." – P1

Hosting curation projects on GitHub can encourage contributions. GitHub has a lot of collaborative features. It is a common practice on GitHub for users to contribute to other projects Dabbish et al. (2012); Marlow et al. (2013); Wu et al. (2014). P12 put "GitHub can target at the right audience, and contributing is encouraged more ...". P8 claimed that "... enable other people to (freely) contribute to it is very important to me (and I think other curators also feel the same) so a Git hosting site is ideal."

Participants also believed that other people on GitHub might have more experience and knowledge than they can, and others can contribute to what they do not understand.

> "The main reason is collaboration...I may have some resources but other people may have even better stuffs or ideas to share." – P3

The Use of Curated Resources

Curated resources are useful for software developers to supporting their work, learning a new topic, as well as communicating with others.

Supporting Work

Software developers rely on others' work to accomplish their own projects. Participants of this study report that they use different curated lists, including their own, as bookmarks or references to quickly locate the resources they need and avoid repeated search efforts.

> "Before making the repo I had to do research each time I needed a (resources). Now that I have a list. I just refer back to it when needed. It serves as a good toolkit for future projects." – P12

> "I recently have created a Python repository and since I was not used with Python at all, I used awesome-python to know some libraries recommended by the community." – P11

In addition to supporting their work, participants also use the curated repository to keep track of high-quality resources in case they might need them in the future. For example,

> "If I used it or I'm planning to use it, I'll add it there. If the resource is well written with tests and should be considered while selecting specific category, I'll add it too... but also I add (resources) that I checked already and found it interesting for the future projects." – P16

Learning a New Topic

When first encountering some new topics, software developers often find themselves overwhelmed. The complex information scope in software developers' community makes it hard for software developers to start development tasks quickly. For example, P6 report that "when we start to learn new thing, there are many things, we cannot know what should to spend time on."

A curated list of relevant topic provides them with a perfect starting point, which serves as a centralized resource repository where software developers know that they can find high-quality resources and start learning the topic.

> "... I'm an iOS engineer. But someday I like to learn Ruby, I just go to awesome-ruby and pick some recourse for beginner. Googling is not going to help us like that." – P6

> "So say if I starting to learn a new tool and need to get started quickly. I might go to the main awesome list and search for it." – P5

In this way, curated lists help them stay aware of the scope of an individual topic, and allow them to locate high-quality resources for learning.

Communication

Communication is an essential part of software developers' community that transfers knowledge between stakeholders and facilities learning, coordination, and collaboration (Storey et al., 2014). Curation serves important communication purposes, including reducing communication cost as well as sharing knowledge.

First, curation repositories centralize a number of software development related resources in a single web address, which can be easily shared among software developers.

> "I'm relative active in the meetup community in (P14's location). Talking to people, there is always a lot of talk about what makes a good (P14's curation topic). I created list so that I can point to other easily... I refer a lot of people to the list who are looking at improving their (P14's curation topic)." – P14

Second, curation repositories preserve resources for software developers to share and transfer knowledge. One can share the content of resources with others in the current time, as put by P7:

> "... I sometimes encounter people who've watched (P7's curation repository) and didn't really like them, but my hunch is they haven't seen the great ones, so I send them to check out my list to see if I can convince them otherwise..." – P7

And one can preserve the resources in case it might be needed for sharing in the future. For example, P13 used curation repository as a means to transfer knowledge to upcoming people who would join his team.

> "It helps us in ensuring that the knowledge doesn't get lost when people graduate or leave the team ... It is helpful when new people join the team, we need to assign them material to study ... and we can simply point to the repository" – P13

Limitations of Curation on GitHub

Curation repositories recently emerged and appropriated a platform designed for collaborative coding purposes. It remains restricted by its immature nature and limited supporting features on GitHub. In this section, we summarize constraints that our participants have encountered when engaging with curation on GitHub.

Immature Structure and Format of Current Curated Lists

The README.md file on GitHub aims to include an introduction to each project. The current curation practices mainly rely on that single README.md file to list all curated resources. Sometimes a list may grow excessively long. Participants complained that "resources are not searchable (when on a list)" (P4), and it is cumbersome for them to navigate through a long list:

> "The only thing sometimes that nags me is that some of them are very long, which in some sense defeats the purpose." – P5

In a case, where a curated list is too long, P6 created a shorter version of the same topic, just for selecting resources that are most important ones to him:

> "there is another remote list...lot of stars, around 5k or more, but I find it that there are lots of resources, then when I look into, I'm scared of. Then I want to create my own list, just something I think useful for most." – P6

In addition, as noted in Fig 1, each curated item usually has a brief description, which sometimes can be incorrect, inaccurate, or misleading:

> "Bad description doesn't allow finding the required resource." – P16

Further, although curated list can be a collaborative efforts (i.e., multiple people suggest adding, deleting, or updating entries), there is no intuitive way for audience to express their opinions towards existing ones. One participant suggests including a rating system in the curated list to help audience filter resources:

> "... maybe it would be better we could Like/Dislike the resources ...sometimes the resources are sorted by name when popularity would be a better option... something like this would give us an overview of how much important some entry in a list is for the community." – P11

Excessive Efforts to Filter and Maintain Resources

Curation happens in a complex information space, where an enormous number of resources are distributed over a variety of services. It requires a lot of time and effort to navigate in the information space, and to filter a handful of good resources. P14 emphasized the time constraints for curation:

> "Time. Time is hard...Digging through all of these resources takes time, and I'm usually pretty time constrained." (P14).

In addition, because software industry is changing fast, and resources become outdated in short time, curation repositories required an extra amount of efforts to maintain - get rid of the outdated resources, and to add most update-to-date resources. P16 reported that one drawback of the current curation practices is that curated resources have "no quality update."

Difficulties for Marketing

Although GitHub contains a vast and relevant user base, it does not provide mechanisms for a repository owner to distribute the list directly to the relevant audience. Our participants expressed that it was hard for them to target their repositories to users who are interested in the curation topic they created. For example, P10 conveyed his desire to attract more contributors:

> "the only drawback is the lack of pull requests. I want more... (I want to) discover datasets I missed." – P10

And P4 found that it is demanding to reach out to both potential collaborators and consumers:

> "While it's easy to host a project on github, you still need to put effort into marketing it, so you get other people contributing or finding it." – P4

Unlike social media services, such as Facebook, which curates personalized contents for each user, GitHub only contains technical features to allow users to find the information that they want on their own. If GitHub users are not aware of the existence such kind of curation projects, it would be difficult for them to find these resources in the first place. Therefore, admitting curation practices are embedded in the context of abundant potential collaborators and consumers, it currently still lacks mechanisms and features for effective marketing to potentially interested party.

The above mentioned limitations of curation practices call for future curation hosting services improvement.

DISCUSSION

Our results and analysis present an in-depth view of curation practices on GitHub. In this section, we generalize main findings of the results and discuss design suggestions for future improvement.

Curation as a Communication Channel to Strengthen Software Developers' Community

Storey et al. (2014) differentiate four types of knowledge that are communicated in software developers' community: 1) knowledge in people, 2) knowledge in software artifacts, 3) knowledge socially constructed in a community, and 4) knowledge about developers in social networks. The emerging of curation on GitHub implies that GitHub enhances the third one: communicating knowledge that is socially generated and maintained.

Software developers use a variety of communication channels (e.g., usnet, blogging, social news sites, microblogging, etc.) to exchange the abovementioned knowledge (Storey et al., 2014). Storey et al. (2014) considered GitHub as a platform for software developers to communicate knowledge in software projects artifacts, and they did not see it as a media for communicating socially generated knowledge. In this study, we found that curation projects have transformed GitHub repositories into socially generated knowledge repositories which enabled developers to easily communicate relevant quality resource, learn new topics, and support specific work tasks.

Onboarding new members and educating existing members are essential functions for communities of practice to sustain and grow (Lave and Wenger, 1991; Wenger, 1998; Wenger and Snyder, 2000). Curation repositories on GitHub reflect these core utilities of communities of practices. Software developers exhibit a great passion for technology and learning in online environment (Singer et al., 2013). Curation repositories centralized peer-reviewed resources for guiding one's learning towards a certain topic. It is likely to reduce the amount of efforts the members of the community spend on locating the resources individually and separately, and the quality of the resources potentially make the learning more efficient.

Appropriating GitHub for curation practices are advantageous for software developers' community. Software development related resources are changing rapidly, and software developers usually rely on a number of services and channels, such as Stack Overflow and Twitter, to keep them up-to-date with the trend (Storey et al., 2014). Curation on GitHub provides a collaborative mechanism to organize and maintain resources of interest systematically. It creates a reliable information source to steadily onboarding new members and educating existing members, which in turn helps maintain and grow the community as a whole.

Curation repositories on GitHub evolving as communication tools show the flexibility and reconfigurability of GitHub. With simple appropriation, it becomes a favorite tool intended for another purpose in software developers' community. Such reconfigurability may lead to other practices besides curation, which in turn further benefit software developers' community. For example, GitHub users also started to appropriate GitHub to write and publish software development related books [7], which accept community suggestions as well as changes. Also, software developers start to share training materials for others to discuss related matters as well as retrieving improvements [8].

In general, software developers communicate a variety kinds of knowledge within the community (Storey et al., 2014). Curation repositories as a means of communication can help the community grow because it onboards new members as well as educates current members. Appropriating GitHub for

[7] `https://github.com/getify/Functional-Light-JS/`

[8] `https://github.com/kentcdodds/es6-workshop`

curation demonstrate the reconfigurability of GitHub features, which can be utilized in other ways to strengthen the community further.

Curators as Brokers: Bridging Software Developer's Community

Curators, i.e. software developers who curate resources and share them on GitHub, are similar to technology stewards, who adopt, adapt, and appropriate technologies to satisfy a community's emerging needs (Wenger et al., 2009). Technology stewards serve an important role in communities of practice, where they help bridge information among different groups of people within a community, acting as brokers. In this sense, curators on GitHub serve the broker role in software developers' community.

Curators and brokers are similar regarding the functionality they played in a community – they connect people with one another and control the information flow. Curators on GitHub relate the resource providers, i.e. software developers who develop tools, packages, frameworks, etc., with resource consumers, i.e. software developers who need to learn or work with tools, packages, frameworks, etc. What's unique in GitHub context is that curators don't have to connect to each person of the different groups socially. Instead, they can focus on creating and maintaining contents of the curation repository, and individuals who have knowledge of the topic and people who consume knowledge in the matter will come voluntarily. In this sense, curators play the brokerage role by mediating the communication between different groups of software developers through artifacts, i.e. curation repositories.

The broker role of curators in software developers' community raises interesting questions of their professional trajectory. Brokers usually take advantages of this unique position, bargain for better terms, and move to a better position after a certain time of brokering (Burt, 2005; Van Liere, 2010). However, the past studies usually focus on the cooperate environment, and the brokerage happens by establishing social connections, which leads to social capital gains (Burt, 2004, 2005). To contrast, the new role of curators in software developers' community happens in online social networks rather than the cooperate environment, and brokering information through an artifact, which is unlikely to lead to gaining much social capital. Whether and how curation practice scan help curators' professional development will be an exciting area for the future research.

The Curator Motivations

Altruism is a typical drive for open source participation (Ye and Kishida, 2003; Lakhani and Wolf, 2003), and it motivates software developers to engage in curation practices on GitHub as well. However, altruism has rarely considered as the most important motivations for driving participation in open source projects. Instead, it alone is considered as unsustainable for open source participation (Ye and Kishida, 2003).

Enjoyment-based intrinsic motivations, the primary reason for taking parts in open source projects, were not specifically mentioned as a drive for the participation in our study. Researchers have learned that intrinsic motivations drive software developers to spend more time and efforts on open source projects (Lakhani and Wolf, 2003), and it is the positively reinforced by community recognition (Ye and Kishida, 2003). Therefore, many software developers commit themselves to open source projects for a relatively long time. Whether intrinsic motivations are involved in driving curators and Whether there is a mechanism that regularly feeds curators' motivations to curate are beyond the scope of this study, which requires further investigation.

Design Implications

Our analysis describes GitHub features as a technical infrastructure that meets the needs of curation practices. However, there is still substantial room for curation practices to improve and grow. Our results show that curation on GitHub requires a lot of efforts to filter resources from all kinds of sources and to actively maintain the existing ones. In addition, abundant resources included in one list also creates navigational difficulties. The current conditions could be improved by 1) empowering curation with automated filtering tools, and 2) adding navigational support within a curated list.

Automated tools can reduce the amount of efforts curators need to spend on curating processes. Curators currently do manual selection and evaluation of potential resources as well as eliminating outdated resources. Selections are usually achieved by employing search engines or following recommendations from others. Automated tools can help curators reduced the manual efforts spent on finding and maintaining resource lists. For example, some of our participants manually refer to third party tools to check resources status, such as last-updated-date. An automated tool that checks and filters resources according to query fields can largely diminish the noise and reduce time and efforts to select and evaluate resources.

In addition, an automated tool can also help maintain existing resources, for example it can check whether a software project is still under active development or it is deprecated.

Providing navigational support aims at solving the following issues: 1) lengthy curated list, 2) lacking of search function, and 3) lacking of common themes across different curation repositories. To be more specific, anchored table of contents, which is fixated on the screen, gives readers a clear structure of a document, as well as enables them to jump among sections. It thus makes navigating in a long list easier, alleviating the first issue. Adding a search function within a curation repository, allowing users to query keywords of the curated items, can help users explore and find ideal resources promptly. And templates can provide common structure and themes in different curation repositories. For instance, our participants mentioned that one of the features they want each curation repository to have was to include a beginner's section, where they could easily find out hands-on resources. Curation repositories can adopt a template that includes commonly identified themes, such as beginners' section, so that users will be familiar with the structure of different curation repositories and thus locate resources more efficiently.

LIMITATIONS AND FUTURE RESEARCH

We used “curation”, ”curated list” as keywords to search through repository descriptions and identified repository owners as our potential interviewers. Although this search method can return us the most relevant curation repositories, it may not cover the ones that do not involve such keywords in their descriptions.

Secondly, the owners of the top curation repositories did not respond to our interview invitation. Given those celebrity curators' massive audience base and extensive received contributions & attentions, their motivations and practice might be different from the general curators we focused in this study.

In the future, we intend to learn more feedback from pure curation project consumers who have never built any curation projects. Together with what we have learned from this study, we plan to work on designing and implementing tools to help curators select, evaluate, and maintain resource lists more effectively, and allow users to navigate and retrieve desirable resources more efficiently. We hope to have a complete working prototype shown to the community in the near future. We also would like to explore opportunities for future collaboration with GitHub and curators to conduct large scale field experiments in the context of naturally occurring curation practices.

CONCLUSION

In summary, this study seeks to add to the research literature by providing a greater understanding of the motivations that software developers appropriate GitHub for curation and their experiences with such practice. By conducting in-depth interviews with 16 participants about their curation experiences, we uncovered the motivations and natures of curation practices on a collaborative work production platform. Moreover, we found some limitations of current curation practices and discussed possible opportunities for improvement.

REFERENCES

Bamforth, D. B. (1986). Technological efficiency and tool curation. *American antiquity*, pages 38–50.

Burt, R. S. (2004). Structural holes and good ideas 1. *American journal of sociology*, 110(2):349–399.

Burt, R. S. (2005). *Brokerage and closure: An introduction to social capital*. Oxford university press.

Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1277–1286. ACM.

Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2013). Leveraging transparency. *IEEE software*, 30(1):37–43.

Doll, B. (2013). 10 million repositories. https://github.com/blog/1724-10-million-repositories. Accessed: 2015-12-28.

Duh, K., Hirao, T., Kimura, A., Ishiguro, K., Iwata, T., and Yeung, C.-M. A. (2012). Creating stories: Social curation of twitter messages. In *ICWSM*.

Farooq, U., Kannampallil, T. G., Song, Y., Ganoe, C. H., Carroll, J. M., and Giles, L. (2007). Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 351–360. ACM.

Golder, S. A. and Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *Journal of information science*, 32(2):198–208.

Greene, D., Reid, F., Sheridan, G., and Cunningham, P. (2011). Supporting the curation of twitter user lists. *arXiv preprint arXiv:1110.1349*.

Hars, A. and Ou, S. (2001). Working for free? motivations of participating in open source projects. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pages 9–pp. IEEE.

Higgins, S. (2011). Digital curation: the emergence of a new discipline. *International Journal of Digital Curation*, 6(2):78–88.

Jenkins, H., Purushotma, R., Weigel, M., Clinton, K., and Robison, A. J. (2009). *Confronting the challenges of participatory culture: Media education for the 21st century*. Mit Press.

Lacey, A. and Luff, D. (2001). *Qualitative data analysis*. Trent focus Sheffield.

Lakhani, K. and Wolf, R. G. (2003). Why hackers do what they do: Understanding motivation and effort in free/open source software projects.

Lave, J. and Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2006). Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40. ACM.

Marlow, J., Dabbish, L., and Herbsleb, J. (2013). Impression formation in online peer production: activity traces and personal profiles in github. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 117–128. ACM.

Marshall, M. N. (1996). Sampling for qualitative research. *Family practice*, 13(6):522–526.

Matthews, T., Whittaker, S., Badenes, H., and Smith, B. (2014). Beyond end user content to collaborative knowledge mapping: interrelations among community social tools. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 900–910. ACM.

McDonald, N. and Goggins, S. (2013). Performance and participation in open source software on github. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 139–144. ACM.

Singer, L., Figueira Filho, F., Cleary, B., Treude, C., Storey, M.-A., and Schneider, K. (2013). Mutual assessment in the social programmer ecosystem: an empirical investigation of developer profile aggregators. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 103–116. ACM.

Singer, P., Flöck, F., Meinhart, C., Zeitfogel, E., and Strohmaier, M. (2014). Evolution of reddit: from the front page of the internet to a self-referential community? In *Proceedings of the 23rd International Conference on World Wide Web*, pages 517–522. ACM.

Storey, M.-A., Singer, L., Cleary, B., Figueira Filho, F., and Zagalsky, A. (2014). The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*, pages 100–116. ACM.

Strauss, A. L. (1987). *Qualitative analysis for social scientists*. Cambridge University Press.

Van Liere, D. (2010). How far does a tweet travel?: Information brokers in the twitterverse. In *Proceedings of the International Workshop on Modeling Social Media 2010*, pages 6:1–6:4. ACM.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge university press.

Wenger, E., White, N., and Smith, J. D. (2009). *Digital habitats: Stewarding technology for communities*. CPsquare.

Wenger, E. C. and Snyder, W. M. (2000). Communities of practice: The organizational frontier. *Harvard business review*, 78(1):139–146.

Wu, Y., Kropcznyski, J., Prates, R., and Carroll, J. M. (2015). The rise of curation on github. In *Third AAAI Conference on Human Computation and Crowdsourcing 2015*.

Wu, Y., Kropczynski, J., Shih, P. C., and Carroll, J. M. (2014). Exploring the ecosystem of software developers on github and other platforms. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 265–268. ACM.

Ye, Y. and Kishida, K. (2003). Toward an understanding of the motivation of open source software developers. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 419–429. IEEE.

Zhong, C., Shah, S., Sundaravadivelan, K., and Sastry, N. (2013). Sharing the loves: Understanding the how and why of online content curation. In *Proc. ICWSM 2013*, pages 659–667.