What is the Gist? Understanding the Use of Public Gists on GitHub

Weiliang Wang, Germán Poo-Caamaño, Evan Wilde and Daniel M. German
Department of Computer Science, University of Victoria, Canada.
Email: {weiliang,gpoo,etcwilde,dmg}@uvic.ca

***Abstract*—GitHub is a popular source code hosting site which serves as a collaborative coding platform. The many features of GitHub have greatly facilitated developers' collaboration, communication, and coordination. Gists are one feature of GitHub, which defines them as "a simple way to share snippets and pastes with others." This three-part study explores how users are using Gists. The first part is a quantitative analysis of Gist metadata and contents. The second part investigates the information contained in a Gist: We sampled 750k users and their Gists (totalling 762k Gists), then manually categorized the contents of 398. The third part of the study investigates what users are saying Gists are for by reading the contents of web pages and twitter feeds. The results indicate that Gists are used by a small portion of GitHub users, and those that use them typically only have a few. We found that Gists are usually small and composed of a single file. However, Gists serve a wide variety of uses, from saving snippets of code, to creating reusable components for web pages.**

I. Introduction

GitHub has become one of the most important forges for software development. It provides an environment in which developers can collaboratively develop software using the Git version control system. GitHub enhances Git with features that greatly improve collaboration and communication between developers, including event feeds, pull requests, code reviews, and an issue tracking mechanism[1].

Gist is one of the many features GitHub provides to its users. GitHub defines[2] a Gist as follows:

> *"Gist is a simple way to share snippets and pastes with others. All Gists are Git repositories, so they are automatically versioned, forkable and usable from Git."*

How a technology is supposed to be used may differ from how it is actually used. This is especially true for disruptive technologies where people find innovative uses that were likely not envisioned by the creators. In a recent study [9], we described how GitHub repositories are being used for a variety of purposes, not just software engineering. For example, we found that repositories are being used to share data (the Tate Gallery in London uses GitHub to share the metadata of its entire collection[3]), and as data storage (the Boston Globe uses GitHub to host a historical mirror of its newspaper[4]). Building on this work, we would like to know how users are taking advantage of Gists, and if their use is similar to what GitHub intended, or if they are finding innovative ways to use them.

There has been a recent surge in third-party applications that support the creation and management of Gists. For example: the Chrome App *GistBox*[5] helps users save snippets of Web page text as Gists, as well as organize their collections of Gists; the *sublime-github plugin*[6] allows users to save snippets of code as Gists and share their Gists from within the Sublime editor—*gist.el*[7] is a similar module for Emacs and *gist-it*[8] for Atom; and *jist* is a command-line utility for managing multi-file Gists[9]. This activity seems to imply that Gists are gaining popularity among GitHub users, which motivated us to explore Gists and how developers used them.

In this paper, we present an empirical study of Gists in GitHub. The goal of this study was to understand what Gists are and how they are used. To do this, we attempted to address the following two research questions:

RQ$_1$. What do gists look like?
RQ$_2$. How are users using gists?

To answer *RQ$_1$*, we searched the Web for evidence of how users are describing their use of Gists. This included a search of Websites (including blog posts) and Twitter. To answer *RQ$_2$*, we performed a qualitative and quantitative exploratory study of Gists, including their contents and metadata. For the quantitative part of the study, we sampled 750k GitHub users and identified 762k Gists. For the qualitative part of the study, we performed a manual inspection of a sample of 398 Gists.

II. Related Work and Background

A. GitHub

Among the various distributed version control systems (DVCS) available, Git has gained the most momentum. The Git project began in 2005 as a version control system to coordinate the development of the Linux kernel. Due to its functionality, portability, efficiency, and rich third-party adoption, Git has evolved by "leaps and bounds" [14].

There are many Web-based applications that use Git as a back end to host free and open source projects. These

[1] https://github.com/features
[2] https://gist.github.com/
[3] https://github.com/tategallery/collection
[4] https://github.com/bcomdlc/bcom-homepage-archive
[5] http://www.gistboxapp.com/
[6] https://github.com/bgreenlee/sublime-github
[7] https://github.com/defunkt/gist.el
[8] https://atom.io/packages/gist-it
[9] http://charlesleifer.com/blog/jist-a-command-line-utility-for-managing-multi-file-multi-directory-private-gists/

978-0-7695-5594-2/15 $31.00 © 2015 IEEE
DOI 10.1109/MSR.2015.36

applications provide a convenient way for developers to create repositories, clone existing projects, and commit their contributions [14]. These applications also emphasize the social aspects of software engineering.

With more than 8.5 million users[10], GitHub has become one of the most popular of these applications. GitHub is an environment that combines social networking with distributed version control to enhance communication and coordination among software developers [5].

B. Recent Research on GitHub

The development of GitHub has prompted research from many different angles. GitHub stores data on developers' projects, their contributions to other projects, and their interactions with other developers. Some researchers are trying to help employers by analyzing the profiles and activities of developers on GitHub [2,11,16].

Other researchers have focused on the source code in project repositories. Bissyande et al. [1] took advantage of the rich data on GitHub, using lines of code, development teams, and issues that arose as measurements of popularity, interoperability, and impact of programming languages. Other researchers have tried to discover patterns in how developers asses each other and find collaborators [10,12,13], herd behaviour [3], and the relations between behaviour on GitHub and other Q&A Websites such as StackOverflow[11] [15]. In [9], we empirically analyzed the contents of repositories and discovered that GitHub is used for more than software engineering. While GitHub is meant to be a social coding platform, we discovered that a lot of activity within it is driven by developers who are working on their own. However, we have not found any scientific studies focusing on GitHub Gists.

C. Related Tools

There are other snippet management tools for software developers similar to GitHub's Gists, such as *pastebin*[12] and *snipt*[13]. Their existence seems to imply that the sharing and management of snippets is a growing concern among software developers. However, GitHub is the only one that stores the snippets using version control.

D. The Features of Gists

Gists are small snippets of code or text that are stored using Git. Wikipedia provides a good summary of the benefits of such an integration:[14]

> *"Gist builds upon that idea by adding version control for code snippets, easy forking, and SSL encryption for private pastes. Because each 'Gist' is its own Git repository, multiple code snippets can be contained in a single paste and they and be pushed and pulled using Git. Further, forked code can be pushed back to the original author in the form of a patch, so pastes can become more like mini-projects."*

In addition, GitHub makes it easy to create Gists. It also includes a powerful Web-based editor to modify them—it can completely isolate users who may not want to use Git. It also supports the ability to post comments on a Gist, and provides a Web service to retrieve Gists that typesets the contents so that they are ready to be embedded in a Web page.

III. Research Questions and Methodology

Our study focused on answering the following two research questions

RQ_1. What do gists look like? We surveyed the contents and metadata of Gists to get a picture of how they are used and whether users collaborate around them.

RQ_2. How are users using gists? We searched Web pages (including blogs) and Twitter to discover how users describe their use of Gists. We combined this information with the results of the previous research question to get a full picture of how Gists are being used.

Our methodology can be summarized as a mixed methods approach [6] that combines both quantitative and qualitative analysis on a collection of data. The study was composed of three main components, with the first two being used to answer *RQ_1*, and the third to answer *RQ_2*.

1) A quantitative analysis of Gist metadata and contents, based on the Gists collected from a large portion of GitHub users.
2) A qualitative analysis of a sample of Gists based on a small, random sample of the Gists collected. We inferred the purpose of a Gist by inspecting the contents of each file that composed the Gist.
3) A qualitative analysis of users' comments about Gists, based on search results from Websites and Twitter.

A. Data Set

In July 2014, we downloaded the metadata and contents of a large sample of Gists using the following method. We started by downloading the list of GitHub users from the GHTorrent project [8], which contained 2.9M. GHTorrent contains big part of the activity starting from 2012. We filtered these users to identify true users (GHTorrent contains both organizations and users, see [9] for details); This left us with 2.4M users. From this list of 2.4 million users, we randomly sampled 750k (31%). For each of these 750k users, we downloaded the metadata of all their gists (if they had any). Only 103k users had at least one gist. In total, these 103k users had 762k Gists. We also proceeded to download the first 618k Gists. This data is available in the replication package at http://turingmachine.org/2015/gists.

Table II summarizes the metadata associated with a Gist. As Gists use Git for their underlying storage mechanism, users can fork Gists, and modifications to Gists are recorded as commits. GitHub allows any user to make comments on a Gist.

[10] https://github.com/about/press
[11] http://www.stackoverflow.com/
[12] http://pastebin.com/
[13] https://snipt.net/
[14] http://en.wikipedia.org/wiki/GitHub#Gist

Description	Size
Total population of users	2,407,094
Sampled users	750,000
Users with Gists	103,092
Gists of sampled users	762,034
Gists downloaded	618,393

TABLE I: Sample used in this study.

Description	Size
Gist unique identifier	Unique to all Gists in the system
Description	As described by its owner
Files count	Number of files in Gist
Forks count	Forks of Gist made by other users
Commits count	Number of commits pushed to Gist
Commits history	Number of additions and deletions pushed in every commit
Comments count	Number of comments on Gist by other users
Language	List of languages of each file in Gist
Gist size	
Creation date	
Last modification date	

TABLE II: Description of the Gist metadata.

Gists can be composed of one or more files. Each of the files in a Gist contains metadata, which is described in Table III. The *MIME type* attribute documents the type of file using the MIME notation [7]. We also computed metrics on Gists that consisted of text files: number of lines using the `wc` UNIX command; lines of code per file (for source code files) using SLOCcount[15].

Description	Size
Filename	
MIME type	Type of file using MIME notation
Language	For source code, its programming language (based on its extension)
Size	In bytes
Lines*	Number of lines in Gist
SLOCS*	Number of lines for source code files

TABLE III: Description of the file metadata in Gists. Those marked with * were computed from the downloaded Gists.

B. Manual Analysis of Gists

To understand the content of Gists, we randomly sampled 398 Gists among the ones already downloaded for our quantitative analysis. To abstract recurring patterns, we extracted themes from each Gist's content following Creswell's guidelines [4] for coding. We started reading through the Gists to obtain a general idea of their content and then grouped the contents into categories. Because a Gist is composed of one or more files, we considered two strategies to segment them: analyze the content type of a Gist as a whole, and determine the relationships between the files in a Gist. In the first case, a Gist could be labelled with one or more terms.

Three researchers performed the manual analysis to cross-validate the coding process and minimize potential bias.

C. Analysis of Users' Discussions Regarding Gists

We also considered the information in Web pages and Twitter postings that described how Gists are used. We looked for the most relevant Web pages explaining the use of Gists—either official or unofficial—as well as the most recent comments at the time we were conducting the study.

1) Web Pages: To find relevant Web pages, we queried the Yahoo, Bing, Ask, and Duck Duck Go search engines. We used a script to scrape the first page of links provided for the following queries: *"What is a GitHub Gist"*; *"How do I use GitHub Gists"*; and *"What are GitHub Gists"*. Once we collected the links, we manually read each Web page and made note of the suggested uses.

2) Twitter Postings: To find relevant Twitter postings (colloquially known as *"tweets"*), we performed the query *"GitHub Gists"* using the Twitter Web Search interface. To minimize profile bias on Twitter, we performed the query anonymously, in a private Web browser session with no cookies. We narrowed the search to 6 months, from January 1st to July 31st, 2014, which resulted in a collection 492 tweets. We manually read each tweet and—when appropriate—followed the links pointed there, and made note of the usages suggested.

IV. Results

A. RQ$_1$: What do gists look like?

When considering the results in this section, it is important to keep in mind that this study only pertains to public Gists (we are not able to mine private Gists). GitHub does not impose any restrictions on the number of private Gists a user can have (compared to private repositories which are available only to paying users). Hence, if we repeated this study on private Gists, the results could be different.

1) Users and Their Gists: Gists are used by a small proportion of users. As Table IV shows, 14% of users have at least 1 public Gist. Because the population of users with no public Gists is so large, our remaining analysis will concentrate on users with at least 1 public Gist (103k).

Users	Count	%
Having at least one public Gist	103,092	13.8%
Having no public Gists	645,368	86.2%

TABLE IV: Comparison of users with at least one Gist and those who do not have any.

For users with public Gists, the distribution of the number of public Gists per user is shown in Figure 1. The median number of public Gists is 3 (quartiles 1 & 10). Only 4% of users have 30 or more public Gists.

We hypothesized that the number of Gists a user has depends on their use of GitHub—users with more repositories or commits are more likely to use Gists. For this reason, we calculated the correlation between the number of Gists and other activities; We used the set of all users in the sample, not only those with Gists. The correlation between the number of commits a user has performed and the number of Gists they

[15] http://www.dwheeler.com/sloccount/

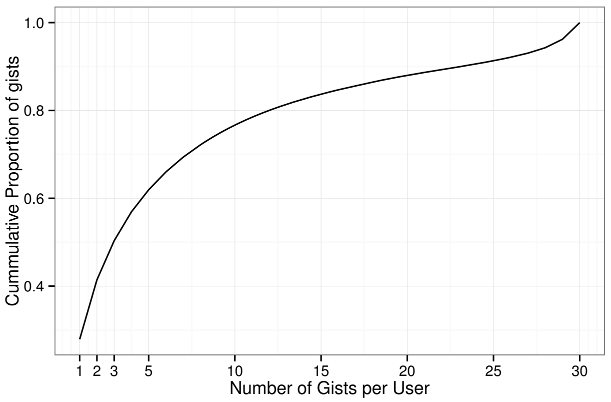

Fig. 1: Accumulated proportion of Gists per user.

have is 0.2, between the number of repos and the number of Gists is 0.31, and between the days since the user registered and the number of Gists they have is 0.37. In all these cases, the p-value was significantly smaller than 0.001. While the correlations are weak, they suggest that as people use GitHub more, they are more likely to use Gists.

2) What Are the Contents of Gists: We quantify the contents of Gists in several ways, starting with files per Gist. In theory, a Gist is a Git repository, and as such, a Gist can contain any number of files. Table V shows the accumulated distribution of files per Gist: 86.8% of Gists contain a single file, and 98.5% of them contain at most 4 files. Very few (0.02%) had more than 10 files.

Files	Gists	%
0	148	0.0%
1	661,565	86.8%
2	53,041	7.0%
3	24,108	3.2%
4	12,588	1.6%
>=5	10,584	1.4%

TABLE V: Breakdown of Gists by the number of files they contain.

GitHub documents the contents of Gist files in two ways: by MIME type, and by the associated programming language or data format (if applicable).

The breakdown by MIME type is depicted in Figure 2. As it can be seen, while the text/plain category dominates (many source code files fall into this category—see below), there is a wide variety of file types. Some of the most common types include images (PNGs are 9.3%, GIFs 0.8%, and JPEGs 0.3%), HTML files (4.7%), JSON data (1.5%), and CSS (2.1%). Some programming languages are identified as MIME types (php, sh, ruby, python, javascript), but many others are not (they are included in the text/plain category).

GitHub identifies the language of the file based on its extension. This includes source code files, data files (such as JSON and XML), and some text files (such as Markdown and HTML). In our sample, 71% of Gists contained at least 1 file classified in this manner, representing 260 different languages. Figure 3 shows the distribution of the top 30 languages. The top 5 programming languages are JavaScript (12.9%), Ruby (11.8%), PHP (10.4%), Python (7.1%), and Shell script (6.3%). The next 4 are markup languages: Markdown (5.9%), HTML (5.7%), XML (4.4%), JSON (2.6%). Note that the remaining 230 languages account for 7.5% (the "Other" category).

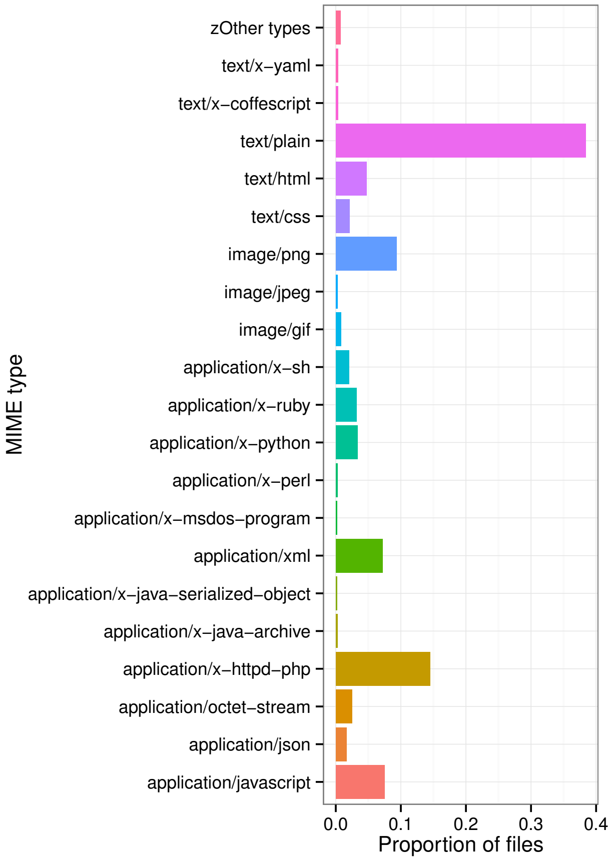

Fig. 2: Proportion of files by MIME type.

When analyzing Gists, we first quantify them by size. Figure 4 shows the distribution of the size of files in Gists using three metrics: bytes, lines, and SLOCs. In terms of bytes, the median size is 920 bytes (quartiles 374 & 2339 bytes). There exist some outliers: 0.06% of files are larger than 1 megabyte. For text files, we counted the number of lines per file: the median number is 22 lines (quartiles 9 & 54 lines). Using SLOCcount, we computed the number of SLOCs in each file. SLOCcount also identified 33 different programming languages (26.4% of files): the median was 18 SLOCs (quartiles 8 & 39).

3) Activity: Because Gists are stored using Git version control, we can trace their evolution (their commits) and the collaboration around them (the number of users who have forked them). In terms of commits, most Gists have very few:

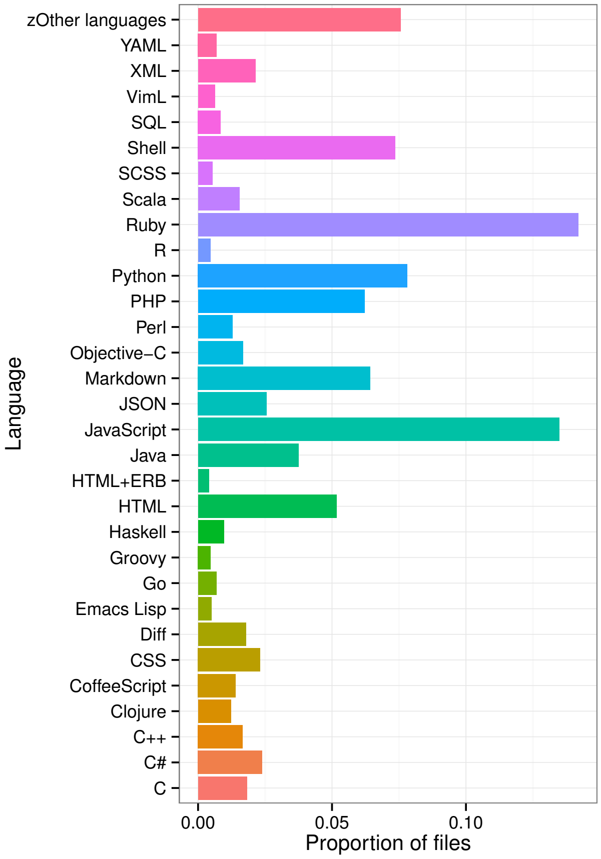

Fig. 3: Proportion of source code files by programing language.

62.9% had a single change, and 92.8% had 2 to 5 commits. The distribution of commits is shown in Figure 5. In terms of forks, only 5.1% of Gists have been forked once, and less than 0.8% have been forked 3 or more times (as shown in Table VI). There are, however, 23 Gists that have been forked more than 100 times. One feature that is different between regular repositories and Gists is that any user can add a comment to a Gist. As with commits and forks, very few Gists have comments (as seen in Table VII): only 6.9% received 1 or more comments.

Number of forks	Gists	%
0	723,833	94.9%
1	28,262	3.7%
2	4,914	0.6%
≥ 3	5,025	0.8%

TABLE VI: Forks per Gist.

4) Manual Analysis of the Contents of Gists: We manually inspected 398 randomly sampled Gists. We segmented the contents of Gists according to two strategies: we analyzed the content of the Gist as a whole, and then analyzed the relationships between the files within the Gist.

Comments per Gist	Gists	%
0	709,098	93.1%
1	33,603	4.4%
2	8,938	1.2%
≥ 3	3726	1.3%

TABLE VII: Comments per Gist.

a) Content Type: We observed that Gists are used for multiple purposes, not only for sharing code snippets. We categorized the Gists as source code (*Code*), any other form of text (*Note*), or both—for those cases when a Gist contains multiple files and at least one file each (*Code* or *Note*).

The breakdown of this first categorization is shown in Table VIII. There is a predominance of source code among Gists, although other types of text (*Note*) cannot be dismissed. The number of Gists that combine both types are low, which is expected given that most Gists contain a single file (see Table V). There are, however, 5 Gists that were classified differently by each researcher (*Not classified*).

Content type	Count	%
Code	290	72.9%
Note	92	23.1%
Both	11	2.8%
Not classified	5	1.3%

TABLE VIII: Breakdown of Gists by major categories of content type.

As seen in Table VIII, code is the most prominent use of Gists. However, we also found other uses, such as system configuration information, sharing public cryptographic keys, generic letters, or even a menu for—apparently—a restaurant. One Gist was written entirely in Japanese, for which we used an online translation tool to understand its content. Thus, GitHub Gists are not limited to western languages.

At the same time, we determined additional categories to better describe the purpose of a Gist given its content. The resulting categories are (in alphabetical order):

- Blog: Technical content in narrative format.
- Class: Definition of one class or module (source code).
- Command: A short command to be run in a shell (source code).
- Configuration: Configuration files used to build code, or for any other purpose.
- Data: Data stored in JSON, CSV, or other format.
- Diff: Differences between files or different versions of the same file.
- Documentation: Explanatory text about a piece of code or technology.
- Function: Definition of one or several functions (source code).
- Fragment: Partial piece of code (that is not a function nor a class), text, or command.
- Log: System log files, output of a program, and/or error messages.
- Non-technical: Notes without any technical content.

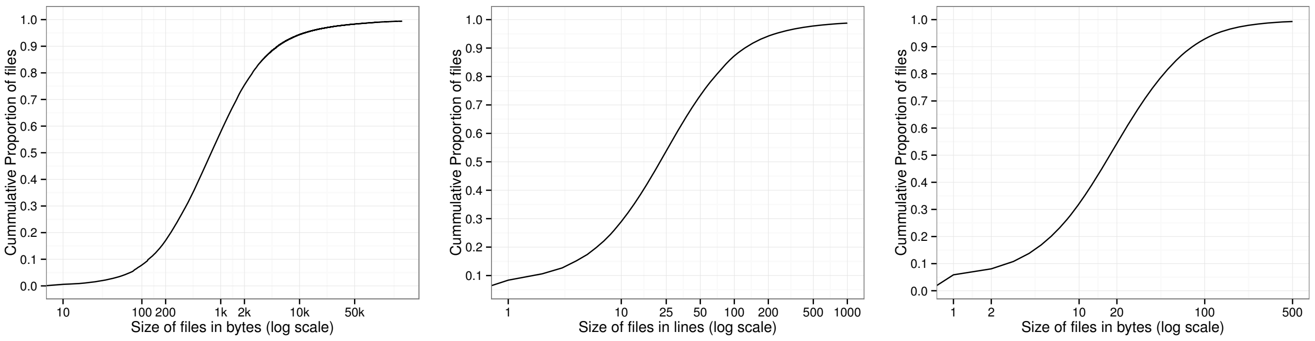

Fig. 4: Size of files in Gists in bytes, lines, and SLOCs

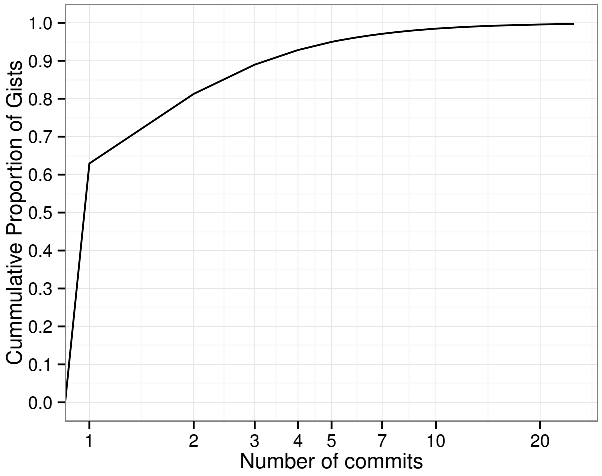

Fig. 5: Accumulated distribution of the number of commits per Gist.

- Template: Coding example or text with patterns.
- Test: Code or text used to test a program or system.

During the categorization process, we observed that the MIME type of a file might not be enough to understand a Gist. For example, the content of an HTML page could contain documentation, embedded JavaScript code, or be used as input for another program. We also observed that a Gist can be segmented in multiple categories, especially the Gists composed of multiple files.

Table IX contains the distribution of the more detailed categories based on the content of Gists. As a Gist can be classified in multiple categories, the sum of them do not represent 100% of Gists. Morever, we observed that 81 (20.35%) of the Gists were complete scripts, modules or programs. Hence, they are not represented in any of the categories.

b) When a Gist Has More than One File, They Are Related: A Gist is composed of one or more files. When a Gist has multiple files, we observed that they appear to be related. We used the following labels to categorize these relationships:

Content type	Count	%
Function	75	18.8%
Fragment	64	16.1%
Data	43	10.8%
Class	43	10.8%
Log	37	9.3%
Configuration	37	9.3%
Test	34	8.5%
Command	27	6.8%
Documentation	22	5.5%
Template	20	5.0%
Blog	14	3.5%
Non-technical	8	2.0%
Diff	7	1.8%

TABLE IX: Distribution of Gists by specific content type.

- Attachment: One file contains configuration or information of another file.
- Generation: One file is the input or output of another file.
- Reference: One file refers to another file, calls functions or methods defined in another file.
- Independent: Files in the Gist are independent or the same file is repeated.
- Single File: The Gist contains only one file.
- Test: One file is used to test the code of another file.

As shown in Table X, most sampled Gists contained only one file (this is consistent with the results of the quantitative analysis). For multi-file Gists, they are spread uniformly among the different categories. Aside from “Single file”, only “Independent” does not reflect an actual dependency between files.

Relationship between files of each Gist	Count	%
Single file	341	85.7%
Independent	19	4.8%
Reference	16	4.0%
Generation	11	2.8%
Test	5	1.3%
Not classified	4	1.0%
Attachment	2	0.5%

TABLE X: Distribution of Gists by relationship between filess

B. RQ_2 How are users using gists?

As described in Section III, we answered this question using two different sources of information: Websites and Twitter messages. Where appropriate, we provide quotations from the sampled Websites and tweets to illustrate our findings.

1) Suggested Uses on the Web: To learn more about how people are using Gists, we searched for "How do I use GitHub Gists", "What are GitHub Gists for", and "What is a GitHub Gist" using the Yahoo, Ask, Bing, and Duck Duck Go search engines. Then we made note of the uses discussed on the first page of results returned for each search.

We found that there were two prominent usage categories for Gists: official and unofficial. In the official usage category, the community uses Gists for code sharing, syntax highlighting and embedding in forums, and for simply storing snippets of code. GeoJSON map rendering is the only official usage that was unmentioned in the community.

Overall, we observed that Gists are suggested as a solution in cases where a full Git repository would be unnecessary.

a) Storing Code: Gists are intended as places to store snippets of code or other small pieces of information.

> *"Instead of creating a complete repository for only 1 or 2 files all the time, just add them to a Gist."*[16]

b) Sharing Code: GitHub used to allow private gists, which were SSL encrypted to protect the contents; however it appears that this is no longer an available feature. "Private" Gists have been replaced with "secret" Gists. Secret Gists have an obfuscated URL but are otherwise visible to the public—simply providing the URL of the Gist to team members gives them access to a wiki-like code sample or snippet. By having all gists visible by the public, it encourages sharing of code. For the purpose of this paper, "private" and "secret" gists are interchangeable.

> *"Gists are a great way to share your work. You can share single files, parts of files, or full applications . . . Every Gist is a Git repository, which means that it can be forked, cloned, and manipulated in every way."*[17]

c) Embedding Code: Gists allow users to embed the contents into blogs, forums, or any text field that supports JavaScript. This allows people to focus on their message rather than on the process of formatting the code in HTML.

> *"You can embed any Gist in your Web pages with a line of JavaScript code."*[18]

d) To-Do Lists: Gists can serve as to-do lists, using the markdown rendering. The version control system records the times when tasks are completed, adding functionality at no cost.

> *"[To-do lists] help me stay organized, prioritize my day, and add structure to an otherwise chaotic schedule. I recently discovered what appears to be the best yet simplest way to keep a to-do list: a GitHub Gist."*[19]

e) Web Hosting: The community has developed third-party applications that can render any HTML code stored in a Gist, making Gists an effective single-page Website.

> *"This [Website] is a simple viewer for code examples hosted on GitHub Gist. Code up an example using Gist, and then point people here to view the example and the source code, live!"*[20]
> *"You can write your HTML, CSS and JavaScript code in plain text, save the Gist as index.html and then use bl.ocks.org to serve the rendered version of that HTML Web page as it should appear in the browser."*[21]

f) Editing Text: The markdown rendering in Gists makes them a simple Web text editor.

2) Uses of Gists Reported on Twitter: We analyzed 6 months of Twitter messages that contained information on GitHub and Gists. We filtered the postings that described or linked usages of Gists, then we categorize them. The resulting categories are described below.

a) Sharing Content: GitHub Gists is used to share snippets of code or any form of text. Although it is one of the purposes featured by GitHub regarding to Gists, we observed people advertising Gists frequently in Twitter.

> *"Gists - https://gist.github.com/ Gist is a simple way to share snippets and pastes with others. All gists are Git repositories."*[22]

We also observed that a collection of related Gists can become a repository over time. For example, we observed 8 tweets[23] linking to a repository containing an introduction to Monad Transformers[24]. The repository was originally a collection of more than 30 Gists with snippets that evolved into a repository[25].

b) Saving Learning Outcomes: Gists can be used to aggregate disperse information technical or record technical tips and learning outcomes tips. For example, a tweet[26] linked to a Gist[27] containing a compilation of business models used by different Internet companies. The Gist is a compilation list of information gathered from a discussion in Hacker News—a social news web site. Although the tweet looks like a call for moderation on how people used Gists, the linked tweet had attracted a considerable attention. At the time of this study, it had received 24 comments, 35 revisions, 201 forks,

[16] https://www.adayinthelifeof.nl/2010/12/26/github-gists-revisioned-code-snippets-for-free/

[17] https://help.github.com/articles/about-gists/

[18] http://www.labnol.org/internet/github-gist-tutorial/28499/

[19] http://lifehacker.com/why-a-github-gist-is-my-favorite-to-do-list-1493063613

[20] http://bl.ocks.org/

[21] http://www.labnol.org/internet/github-gist-tutorial/28499/

[22] https://twitter.com/sstranger/status/493436023729586176' '

[23] E.g. https://twitter.com/philadev/status/490531557816692738

[24] https://github.com/kqr/gists/blob/master/articles/gentle-introduction-monad-transformers.md

[25] https://github.com/kqr/gists

[26] https://twitter.com/pessimism/status/492255854163689472

[27] https://gist.github.com/ndarville/4295324

and 1,633 stars. In comparison to the other results—seen in sections IV-A2, IV-A3, and IV-A4)—this Gists is an outlier because it is not related to code (23.1%), it has more than 3 comments (1.3%) and more than 3 forks (0.8%).

c) Collaboration: Gists also act as a tool to help people with collaboration. People can put their work in a private Gist and every member can commit to it, as is inferred from the following tweet:

> *"Every once in a while I think I wish I could "draft" a Pull-request, issue, or comment on GitHub, then I remember that private Gists exist."*[28]

d) Embedding Content in Blogging Platforms: One of the largest blogging Websites, WordPress, supports the embedding of Gists[29]. Many other blogging platforms, such as Medium, also support Gist embedding.[30]

> *"Thinking of migrating all of the code for my blog posts into @GitHub Gists like [url] - Would that be valuable to you?"*[31]

e) Version-controlled Lists: Some users have come up with non-trival ways to make full use of Gists. One good example is a popular blog being widely tweeted on Twitter. Authored by Carl Sednaoui[32], it teaches people how to maintain a to-do list using a private GitHub Gist.

> *"GitHub Gists are a great way to keep version-controlled lists (in this case, US states I've visited)... "*[33]

V. Discussion of Results

With Gists, GitHub provides a simple way to create and version small files. Among the users we sampled, only 1 in 8 had public Gists, and most of these users had very few in total (the median number is 3). Our results show that someone's use of Gists is correlated with the length of time they have used GitHub. As such, we expect that the use of Gists increases significantly over time. It is also likely that there is a critical mass effect: As more people use and talk about Gists, others will follow suit.

As GitHub expected, most Gists are very small: 86.8% have only 1 file and the median size is 920 bytes (22 lines for text Gists). However, we found that their contents vary widely. While a large proportion contain source code, people are also using Gists for binary files (such as images) and data files (such as XML and JSON). Based on our qualitative analysis and manual sampling of Gists, the following themes have surfaced.

A. Gists Are Mostly Used to Store Source Code but Other Formats Are Frequently Used Too

As expected, Gists are mostly used to store snippets of source code. Our manual analysis showed that they cover a wide range of uses: shell scripts, class templates, complete functions, fragments of functions, etc. Because most do not evolve, we hypothesize that these snippets are being archived for future reference. In addition to source code, Gists also contain other information formats. The Markdown markup language is the fourth most common language, suggesting that storing snippets of text is also an important use of Gists. Similarly, we found that almost 10% of Gists contain images. Gists are also used to store logs, diffs, JSON data, and test data.

B. Gists Are Used to Create Reusable Web Components

GitHub provides a mechanism to dynamically embed Gists into Web pages. When a Gist is embedded, it is nicely typeset (according to the syntax of its language) into HTML. This includes Markdown and Org, both markup languages for text designed to be converted into HTML. When a Gist is rendered, it is bound with a box and text that identifies it as a Gist hosted in GitHub. However, this box can be removed. For example, gist-embed[34] enhances GitHub's rendering of Gists by removing any signs that the content comes from GitHub. Gists can act as dynamic "includes" in Web pages (whether formatted text or source code). Other formats that do not require typesetting (such as images and CSS) are easier to reuse since they do not need to be embedded. For example, if one wants to host an image in GitHub, all that is needed is to create a Gist and then refer to this Gist using the GitHub URL that retrieves the original content (the "raw" Gist). This is likely one of the reasons that we found that almost 10% of Gists are images, and might explain why 8% are JavaScript snippets. Dynamically using Gists in Web pages (either embedded or in raw format) has three advantages:

1) For languages that must be converted into HTML (such as Markdown or source code), it allows authors to ignore the complexities of authoring HTML, and concentrate on creating content.
2) It allows the reuse of components for the Web. The same Gist can be reused multiple times.
3) It isolates the evolution of the component from the use of the component. Because Gists are used dynamically (in raw format, or embedded and rendered) by Web pages, they can be updated without having to change the Web page that uses them.

The evidence we have collected suggests that this is an important use of Gists, especially for Web pages that include source code. Converting source code to a nicely typeset HTML would require the use of extra tools. By hosting the code snipped as a Gist in GitHub, the job of rendering the code into HTML is no longer the responsibility of the user. As a result, it is not only easier to embed source code into a Web

[28] https://twitter.com/nuclearsandwich/status/249213040610910209

[29] http://crunchify.com/how-to-embed-and-share-github-gists-on-your-wordpress-blog

[30] https://medium.com/the-story/yes-we-get-the-gist-1c2a27cdfc22

[31] https://twitter.com/jessealtman/status/456390107952467968

[32] http://carlsednaoui.com/post/70299468325/the-best-to-do-list-a-private-gist

[33] https://twitter.com/dliggat/status/458090816930848768

[34] https://github.com/blairvanderhoof/gist-embed/blob/master/gist-embed.js

page, but the source code looks good. It is hard to evaluate the impact of this feature, but it is likely this rendering has a direct impact on the readability of Web pages that present source code—compared to simply using a pre-formatted HTML tag (<pre>), for example.

C. Gists Are Used as an Enhanced Online Document Editing System That Adds Version Control Features

With Gists, GitHub provides a lightweight method to edit documents with the full benefits of version control. GitHub isolates the user from the creation and editing of Gists from Git. A user only needs a Web browser to gain all the benefits that Git provides for tracking changes (when a change is made, who made the change, and what the change was). We have found evidence that Gists are used for this purpose, even though it is not common (less than 8% of Gists have more than 6 changes).

D. Users Do Not Collaborate Around Gists

Even though we studied public Gists, we found that they are mostly personal artifacts. Gists are rarely forked, and the majority of Gists never change. Unfortunately, GitHub does not provide statistics regarding how many users visit a Gist, nor when. Therefore, we cannot create a picture of how useful Gists are to those who do not fork them or modify them. It appears as if some Gists serve as an external memory to their owner, a memory that the owner is happy to share with everybody (when the Gist is public).

E. Gists Are a Public Scrapbook

As described above, it is very likely that many public Gists are reusable components, intended to be used by their owner in the creation of Websites. However, it is also likely that many Gists are artifacts that the user would like to share with anybody who finds them useful. In a way Gists are a public online scrapbook where developers can collect small artifacts that they find useful, and that can also be potentially useful to others.

VI. Future Work

This study is exploratory. We are just scratching the surface on what Gists are and how they are used. One aspect that we have not researched is what users think about Gists, and future work should survey and interview users. For example: What motivates a person to create a Gist? What do they use Gists for? What factors determine if a Gist should be public or private? Do they expect their Gists to be used by others? Our study was limited to public Gists. Are private Gists different from public Gists?

Another area that requires future research is exploring how users find and reuse Gists (which they have not authored). GitHub provides a search engine for Gists, hence, it expects users to benefit from the Gists of other users. It could be also interesting to study further why the scripting languages used in Gists outnumber other languages such as Java or C; is sharing snippets easier for scripting languages? is this an indicative of activity on GitHub projects developed in such languages?

We hypothesize that some Gists are created for future use. In this case, the user has considered that the snippet is important enough to be remembered as a Gist. It could be interesting to see if there are some commonalities among the Gists of different users. Are different users storing similar Gists?

When a Gist is meant to be reused in the future, such Gist is a potential reusable component. Gists might provide an interesting view on reuse at a higher level of granularity than libraries. It could be interesting to perform clone detection between Gists and source code in the owners' repositories (and other repositories) to find out how reused a Gist is. This could mean that there are certain functionalities that the user frequently requires.

GitHub is not the only Website that stores snippets of code. We need research that explores other repositories and compares them to the results described herein.

VII. Threats to Validity

We triangulated the results of the quantitative and qualitative analysis to overcome potential threats to validity. The quantitative analysis was performed by two researchers, while the qualitative analysis was performed independently by three researchers on the same data set, which will reduce the likelihood of erroneous results. In this section, we explain how we addressed each threat to validity. We are providing an online package that includes the data we analysed and the results of the manual analysis at http://turingmachine.org/2015/gists.

A. Construct Validity

The manual categorization of Gists may introduce errors into the results. The categorization activity involved having a researcher categorize the Gist by interpreting the contents of the files contained in the Gist. If a researcher misinterpreted the contents, it would introduce errors. To minimize the errors introduced by misinterpretation, three researchers categorized the 398 Gists in the sample. Each researcher followed Creswell's guidelines [4] for coding to minimize the introduction of subjective bias by the researcher.

An unsuitable sample of the Twitter posts (tweets) and Web search engine results can also affect the validity of the results negatively. Twitter returns posts with priority given to more recently posted results—the results of the query depend on the time it was executed. This dependence makes the results transient and likely to change. While search engines are still susceptible to changes over time, they use a score-based search algorithm to return the most relevant pages, making the results less transient. The internal details of each search engine are unknown to us; it is unclear wether the results returned by the search engines are representatives samples of the queries performed. We hope that the qualitative analysis of both data sets, in conjunction with the qualitative and quantitative analysis of Gists, reduces distortion in the results.

B. Internal Validity

GitHub provides secret Gists, which are hidden from search engines and the public forum, but are available to anyone with the Gist identifier. We performed the analysis on data collected from public Gists so our results are limited. The contents of secret Gists may differ from the contents of public Gists.

C. External Validity

This study is exploratory and only applies to Gists in GitHub. While there are other snippet storage sites on the Internet (such as *pastebin* and *snip*), we do not make any claims regarding the generalizability of our results to those other sites.

VIII. Conclusions

In this paper, we conducted an exploratory study of GitHub Gists, quantitatively measuring 762k Gists that belong to 750k users, manually coding the content of hundreds of Gists, and exploring the common Gist usages described in Web pages and on Twitter. Our qualitative analysis allowed us to identify recurring patterns in the data that might be difficult to detect quantitatively.

Our goal was to understand the purpose of Gists and how they are used. We summarize our results below.

RQ_1. What do Gists look like? Usually a Gist is a small snippet of source code. Although we found Gists that did not contain source code, those were less frequent (23.1% in contrast to 72.9% of code). In most cases, Gists are composed of one file whose size is relatively small. We also found that Gists are written in many different programming languages, with JavaScript and Ruby being among the most popular.

RQ_2. How are users using Gists? The usage of Gists goes beyond the official purpose promoted by GitHub. GitHub describes Gists as a way to share source code and to embed source code into external services such blogs or forums. However, we found that Gists are also used to maintain online notes with the full benefits of version control. We also found an incipient set of tools that help users manage their Gists; We expect the number of such tools to grow as Gists become more popular.

References

[1] T.F. Bissyande, F. Thung, D. Lo, Lingxiao Jiang, and L. Reveillere. Popularity, interoperability, and impact of programming languages in 100,000 open source projects. In *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*, pages 303–312, July 2013.

[2] A. Capiluppi, A. Serebrenik, and L. Singer. Assessing technical candidates on the social web. *Software, IEEE*, 30(1):45–51, Jan 2013.

[3] Joohee Choi, Junghong Choi, Jae Yun Moon, Jungpil Hahn, and Jinwoo Kim. Herding in open source software development: An exploratory study. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work Companion*, CSCW '13, pages 129–134, New York, NY, USA, 2013. ACM.

[4] John W Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, volume 2. Sage Publications, 2009.

[5] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Leveraging transparency. *Software, IEEE*, 30(1):37–43, Jan 2013.

[6] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting Empirical Methods for Software Engineering Research. In *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer London, 2008.

[7] N. Freed, J. Klensin, and T. Hansen. Request for Comments: 6838 Media Type Specifications and Registration Procedures. Internet Engineering Task Force (IETF) http://tools.ietf.org/html/rfc6838, 2015.

[8] Georgios Gousios. The GHTorrent dataset and tool suite. In *MSR '13: Proceedings of the 10th Working Conference on Mining Software Repositories*, may 2013. Best data showcase paper award.

[9] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. The promises and perils of mining GitHub. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 92–101, New York, NY, USA, 2014. ACM.

[10] Anirban Majumder, Samik Datta, and K.V.M. Naidu. Capacitated team formation problem on social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1005–1013, New York, NY, USA, 2012. ACM.

[11] Jennifer Marlow and Laura Dabbish. Activity traces and signals in software developer recruitment and hiring. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 145–156, New York, NY, USA, 2013. ACM.

[12] Jennifer Marlow, Laura Dabbish, and Jim Herbsleb. Impression formation in online peer production: Activity traces and personal profiles in GitHub. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 117–128, New York, NY, USA, 2013. ACM.

[13] Leif Singer, Fernando Figueira Filho, Brendan Cleary, Christoph Treude, Margaret-Anne Storey, and Kurt Schneider. 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*, CSCW '13, pages 103–116, New York, NY, USA, 2013. ACM.

[14] D. Spinellis. Git. *Software, IEEE*, 29(3):100–101, May 2012.

[15] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. Stackoverflow and GitHub: Associations between software development and crowdsourced knowledge. In *Social Computing (SocialCom), 2013 International Conference on*, pages 188–195, 2013.

[16] Rahul Venkataramani, Atul Gupta, Allahbaksh Asadullah, Basavaraju Muddu, and Vasudev Bhat. Discovery of technical expertise from open source code repositories. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 97–98, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.