# The Diffusion of Pastebin Tools to Enhance Communication in FLOSS Mailing Lists

Megan Squire(✉) and Amber K. Smith

Elon University, Elon, NC, USA
{msquire,asmith90}@elon.edu

**Abstract.** This paper describes how software developers who use mailing lists to communicate reacted and adjusted to a new supplementary collaboration tool, called a pastebin service. Using publicly-available archives of 8800 mailing lists, we examine the adoption of the pastebin tool by software developers and compare it to the model presented in Diffusion of Innovation (DoI) theory. We then compare the rate at which software developers decided whether to accept or reject the new pastebin tools. We find that the overall rate of pastebin adoption follows the S-curve predicted by classic DoI theory. We then compare the individual pastebin services and their rates of adoption, as well as the reaction of different communities to the new tools and the various rationales for accepting or rejecting them.

**Keywords:** Open source · Pastebin · Email · Diffusion of innovations · Software development

## 1 Introduction

Software developers working in distributed teams, such as on free, libre, and open source software (FLOSS) projects, have historically used digital tools to communicate with each other about bugs, features, and decisions related to the project. Developers on these types of projects are also often geographically and temporally distributed, making the use of digital media a requirement for communication. Traditionally, the most common digital tool for software development communication has been the email mailing list. Many of these email mailing lists are publicly-viewable and archived for the long term, since FLOSS development relies on a certain level of openness in participation, transparency in decision making, and institutional memory. Apache Software Foundation projects, for example, are required to conduct all official project business on the mailing list (e.g. [1] [2]). Software developers can use email mailing lists to send each other long text artifacts for review, such as bug reports, code snippets, error logs, and the like.

However, when compared to newer social media web sites, email mailing lists can seem simplistic. As [3] explains, over time, more software development will be conducted by "social programmers" using web sites and apps designed for sharing artifacts, and mailing lists risk obsolescence. For example, while mailing lists can be expressed as a type of primitive social network [4], they are not nearly as expressive of social relationships as are microblogging services like Twitter or the pull request system of Github. For reputation management, email lacks the badging and voting


E. Damiani et al. (Eds.): OSS 2015, IFIP AICT 451, pp. 45–57, 2015.
DOI: 10.1007/978-3-319-17837-0_5

features of Q-and-A web sites like Stack Overflow. For collaborative editing, mailing lists are not as easy-to-use as a wiki or a shared code editor.

And yet, email persists as a pillar of "social programming". In the mid-2000s a class of web site was created to facilitate sharing source code via email. A site like this is called generically a "pastebin", after Pastebin.com, one of the first such sites. A pastebin is a web site that allows a user to paste in text and receive back a permanent short URL to the text that was pasted. Some pastebins even include syntax highlighting for common programming languages. This can be an appealing advantage when constructing an email with source code in it, or multiple attached log files, or long error logs, or complicated bug reports. Developers using a pastebin advocate that it enhances the utility of the email mailing list as a social communication tool: it makes sharing text easier and reading more efficient.

For this paper, our research questions center around diffusion of these pastebin tools on software development mailing lists, as follows:

- RQ1: What is the rate of diffusion for pastebins among FLOSS developers using mailing lists?
- RQ2: Does the rate of diffusion change between different variants of the innovation?
- RQ3: What are the stated reasons for and against adopting this innovation?

By studying the rates of diffusion of this pastebin tool, we can better understand how contemporary distributed software development is done, and whether certain tools designed to facilitate social programming on older communication tools will be adopted or not.

To answer these questions, Section 2 gives some background on the pastebin innovation itself and we review the tenets of classical Diffusion of Innovation (DoI) theory. In Section 3 we present our methods for measuring the diffusion of this innovation in the FLOSS developer community, specifically in its email mailing lists. In Section 4, we review the results of this analysis and present our findings for how and why the innovation diffused, including differences across tools. In Section 5 we discuss the implication of our method and analysis on our research questions. In Section 6 we explain limitations of our method and in Section 7 we make recommendations for future study.

## 2 Background

### 2.1 Pastebins

As the frequently asked questions (FAQ) document of one popular pastebin tool explains, the website "is mainly used by programmers to store pieces of sources [sic] code or configuration information, but anyone is more than welcome to paste any type of text. The idea behind the site is to make it more convenient for people to share large amounts of text online." [5] Examples of pastebins include: Pastebin.com, Github Gists, and Paste.org.

The innovation of a plain pastebin represents the unification of several previous ideas: a pastebin at this level is simply a very quick way to publish a text document on

the web and give it a shortened URL suitable for sharing. Pastebin.com requires no authentication (although that option is available), and the URLs can be set to expire in increments ranging from minutes to "never". The goal is to simplify the publication process from a complicated one involving file transfer and permissions-setting, to one involving only pasting into a web form.

More recently, there is also a class of web sites called an online IDE (integrated development environments) which not only allow the developer to paste in code and provides a link back, but also allows this code to be modified and run in the browser. Examples of online IDEs include CodePen, jsFiddle, and JS Bin (for testing JavaScript code), SQL Fiddle (for testing SQL), PhpFiddle (for testing PHP code), and CodePad (supports a variety of languages).

We should note that because of the anonymity and convenience provided by pastebin sites (and even online IDEs), they have also been used for non-software related purposes, including some illegal or of questionable legality (for example sharing of password lists, posting stolen credit card numbers, high volumes of spam link farming, and the like). See [6] for a description and timeline of some of the more notable illegal activities on Pastebin.

Another important aspect of all pastebin sites is the longevity of the provided URL. Each pasted document is initially given a unique URL, but if the document is allowed to expire, this URL will become a dead link ("link rot"). Pastebin sites have also gone defunct (perhaps due to their site administrators not expecting the high volume of "alternative" uses of their sites, see [7] for details about the expense involved in eradicating spam on Pastebin.com) and this means any links are also defunct.

### 2.2 Diffusion of Innovations

We are particularly interested in how these pastebin tools diffused into common usage among software developers. How innovations diffuse (diffusion of innovations, or DoI) has a rich literature, stemming from the work of Everett Rogers [8] who laid the foundation for how to systematically study the process that a new idea goes through as it becomes accepted or rejected by a community. Among Rogers' contributions was to document the "S-curve" that innovations typically go through on their way to becoming accepted or diffused. If we plot time on an X-axis and usage/acceptance of an innovation on the Y-axis, the typical path of an innovation over time will look something like the letter "S". See Figure 1 for an example. The steepness or shallowness of the S will be interesting to the diffusion researcher (as will the lack of an "S" shape, if the innovation was not successful in diffusing at all).

DoI research also tends to be concerned with the processes that led to that particular rate of diffusion, including characteristics of the community (or individuals) that would affect the diffusion, such as the compatibility of the innovation with the community's values or the effectiveness of a change agent to champion the innovation. Diffusion can be quite complex, especially with technologies that are themselves complex or networked systems [9]. Entities (governments, companies or individuals) with a stake in getting their particular innovation adopted will attempt to understand the myriad variables that can affect the rate or likelihood of diffusion, in order to make that diffusion process more effective.

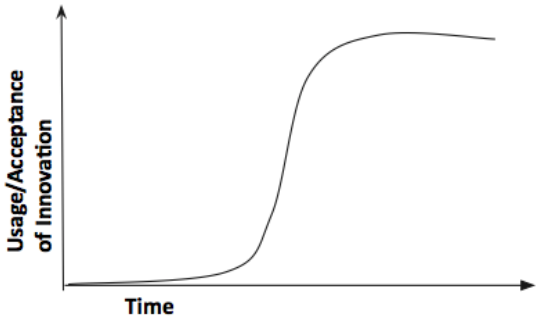


**Fig. 1.** Typical Diffusion of Innovations curve (modeled after Rogers, 1964)

Once diffusion is determined to have occurred, interesting questions may center around the following: describing the initial discovery of the tool by early adopters, the steps taken by early adopters to encourage others to use the tool, the rejection and refusal by some community members to use the tool, and the expectation-setting and rule-setting by the community governing use of the new tool.

For this project, we are primarily interested in doing the foundational work of calculating the rate of diffusion of pastebin tools within the community of social programmers who use mailing lists to communicate. We also take the first steps toward understanding how early adopters encouraged later adopters to use the pastebin innovation.

## 3 Methods

### 3.1 Data Collection

To answer our research questions, we needed first to retrieve the count of times each pastebin tool was mentioned on software development mailing lists over time. The result will be plotted as a rate-of-diffusion curve. To get this data we first identified a source for searching software development mailing lists by keyword. All of our mailing list data came from a publicly-available email aggregation web site called MarkMail.org. MarkMail provides a search interface for approximately 70 million emails from more than 8800 software development mailing lists, in the time period 1992-2014. MarkMail allows broad keyword searches, and also allows searching within typical email headers (e.g. from, subject, and list address).

With MarkMail as a reliable source of email data, we then had to figure out which words were being used to describe pastebin services. The web site Pastebin.com was begun in 2002, and the first mention of either the Pastebin.com web site or the generic noun ("pastebin") on a mailing list was on May 11, 2003 [10]. Though Pastebin.com was probably the first in widespread, public usage [7], there are many more pastebin-style websites in existence now.

The generic noun "pastebin" is still used to refer to both the site type ("Use a pastebin to post your code!") and the actual text having been pasted ("I can't find that pastebin you sent"), but other, newer pastebin tool names are often used generically as well (for example, "I'm going to create a gist" or "Did you get my dpaste?"). We therefore needed to search for full or partial web site URLs (e.g. "pastebin.com", "gist.github.com"), and

the corresponding generic nouns ("send me a pastebin" or "as you can see from my gist"). To answer our first research question about the rate of diffusion we needed to count all mentions of pastebins, whether by URL or by generic noun.

To identify relevant pastebin tools to use as search terms, we constructed a master list of 38 pastebin web sites and online IDEs that are used in software development. We then searched for each tool in MarkMail to see which were most used by software development teams. The results are shown in Table 1.

**Table 1.** Top 12 Most Used Pastebin Websites on MarkMail (2003 - 2014)

| Rank | Pastebin Website | Message Count | Rank | Pastebin Website | Message Count |
|---|---|---|---|---|---|
| 1 | Pastebin.com | 64009 | 7 | Codepad | 847 |
| 2 | Github Gists | 39557 | 8 | Paste.org | 467 |
| 3 | Pastie | 13397 | 9 | Codepen | 224 |
| 4 | jsFiddle | 10536 | 10 | IDEone | 196 |
| 5 | Dpaste | 6740 | 11 | SQLFiddle | 54 |
| 6 | JS Bin | 2416 | 12 | all others | < 50 ea. |

Most of the tools have the same common name as their URL domain (e.g. jsFiddle). However, two of the pastebin tools, Github Gists and Paste.org, were harder to search for as generic nouns since their relevant stem is already a word in common English usage. Table 2 shows the results for paste and gist as generic nouns, versus their URL-specific versions.

**Table 2.** Comparing the message count for 'gist' and 'paste' as words and as URLs

| Original Keyword | Message Count | New Keyword | Message Count |
|---|---|---|---|
| gist | 68706 | gist.github | 39557 |
| paste | 674760 | paste.org | 467 |

To fix these problems, we decided to search for Github Gists using the partial URL gist.github only, and we limited results for the "Paste.org" website to only those messages that included the partial URL paste.org. We recognize that, by doing this, we may have missed some instances of *gist* or *paste* used generically to refer to the pastebin tool. We also implemented some basic data cleaning procedures. One mailing list (com.googlegroups. jquery-br) used the words JS Bin and jsFiddle in nearly every signature line on over 50,000 messages sent. We removed these.

For each of the four most frequently used pastebin sites shown in Table 1 (more than 10,000 mentions), we collected the data for usage over time (message counts by month and year). We wrote scripts to download the counts for each term. The scripts are available on Github for anyone to use [11]. The next section describes our analysis of this data.

### 3.2 Data Analysis

We have created several charts showing the count of messages with each of the top four pastebin sites mentioned. Figure 2 (next page) shows the overall percentage of

messages mentioning any pastebin site from May 2003 until February 2014. The percentage of mail mentioning any pastebin tool ranges from less than 0.1%, but currently hovers around 0.5% of all mail.

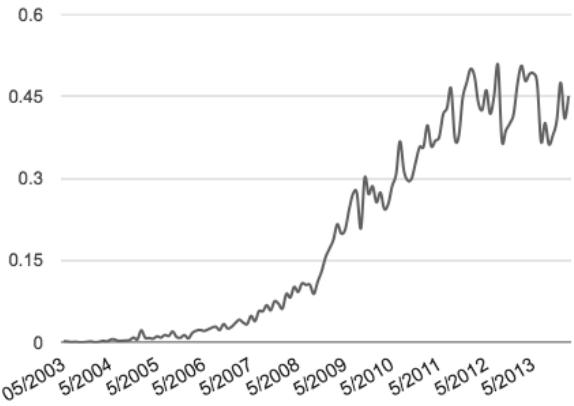


**Fig. 2.** Percent of mailing list messages making reference to any pastebin, 2003 - 2014

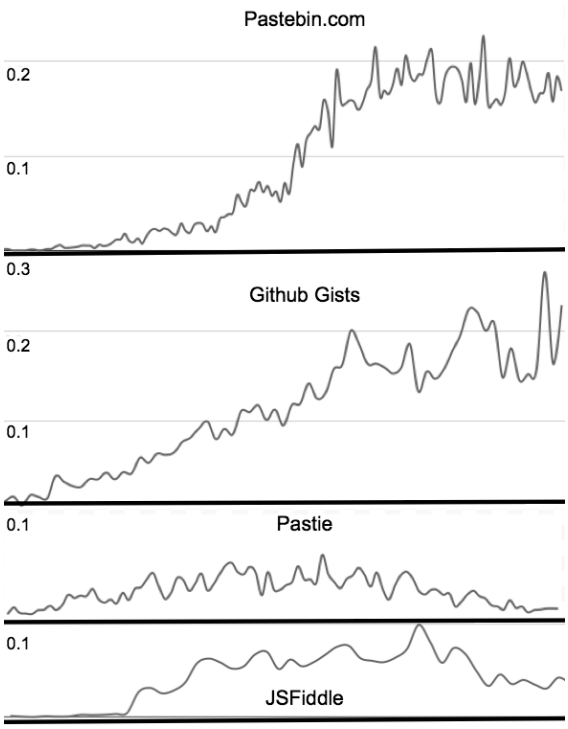


**Fig. 3.** Comparison of pastebin tool references as a percentage of all mail sent, 2003 - 2014

Figure 3 shows each particular diffusion line for the different pastebin tools. The X-axis (time) for each sub-graph in Figure 3 differs by when that pastebin site was created. For example, the first mention of Pastebin.com itself was in May 2003 [10], whereas the first mention of Github Gist was in July 2008 [12]. Each sub-graph in Figure 3 has been smoothed, and each Y-axis shows the keyword as a percent of all mailing list messages in MarkMail (to take into account natural variations in email usage over time by teams in general). These range from <0.1% to nearly 0.3%.

## 4 Discussion

Figure 3 shows some of the same steep increases and "S-curve" that is expected in DoI theory for the largest pastebin sites (pastebin.com, Github Gists), and to a smaller extent with jsFiddle. Pastie is comparatively flat throughout its existence. In this section we describe each of the diffusion curves individually, compare their shapes, and give reasons for the differences.

### 4.1 Pastebin.com Discussion

The curve for diffusion of Pastebin.com is shown in the top bar of Figure 3. The web site for Pastebin.com claims that it was started in 2002, and indeed the first Pastebin.com snapshot [13] on the Wayback Machine [14] was taken on November 23, 2002. But the first mention of Pastebin.com on any of the email mailing lists was six months later on May 11, 2003 [10]. By November of 2003, or one year after its creation, some mailing list users were treating the use of Pastebin.com as a social norm (an expectation for doing business on the mailing list), and some were actually shaming "newbies" for NOT using it [15]:

```
> You need to get a website and post a link to the site instead
of posting a giant email
> with code that's not even indented and probably wrapped in
some places.

Agree with CP here - please don't post large code get some web-
space, or use www.pastebin.com; please don't send attachments;
please don't send your messages priority, request return re-
ceipts; please turn of [sic] vacation auto-responders, please
use plain text, and above all else, please don't top post.
(where's that weekly newbie email?)
```

In Rogers' description of the diffusion of innovations, early adopters often become the champions or advocates of the innovation. However, one thing that is interesting about the message in [15] is that not only does this admonishment happen relatively quickly in the life of the pastebin tool, but this particular list (php-general) is a mailing list specifically designed for all types of users, whereas the original place that the tool was diffused was a developer-specific list. New users ("newbies") are apparently expected to know about the existence of this tool very quickly.

Indeed, the "weekly newbie email" to which the email author refers was indeed sent out just the day before the exchange above [16], but it did not mention

Pastebin.com nor did it even encourage people to point to code on the web by using a URL. In fact, this idea was not officially recommended to newbies on that mailing list until April of 2004, five months after this user was admonished. [17]

Another interesting pattern we see in the diffusion of Pastebin.com is how rapidly "pastebin" becomes a generic noun, and how quickly copycat web sites spring up. The first mention of a copycat web site is about 13 months after Pastebin.com was first created, and about 7 months after the first use of 'Pastebin.com' on mailing lists [18]. The first mention of pastebin as a generic noun is in a February 2004 message [19] describing a new tool called "Trash: a pastebin application written using the Twisted framwork" [sic].

Eventually (certainly by 2009) there is some pushback in at least some portions of the developer community to using a pastebin on mailing lists. The common reasons given are, first, that emails with pastebin links are annoying to read because you have to click a separate link. As [20] begs,

> ...please (please please) don't use pastebin. Just include the output inline in the mail message. It is much easer[sic] to get at then.

Second, developers noticed that the pastebin URLs eventually time out or disappear ("link rot"), making them unreliable for long-term issue tracking. For example the developer in [21] explains,

> ...Pastebin is a useful tool for sharing information between people, but please do not reference pastebin URLs in Jira Tickets. Pastebin entries are not guaranteed to survive for any period of time, and they can and will disappear at some point. This means of [sic] you open a ticket and you put a backtrace in the pastebin, if the pastebin is cleared so is your backtrace.

### 4.2 Github Gists Discussion

Github introduced its Gists service on July 21, 2008, and the first mention of 'gist.github' on a mailing list was two days later [12]. Whereas Pastebin.com took six months to diffuse onto mailing lists, the fact that it took Gists two days to diffuse confirms the fact that pastebins were well-understood by developers at this point. Thus the Gists adoption curve tracks upward more quickly than Pastebin (Figure 3).

However, we still find evidence of some pro-innovation diffusion behaviors inside the developer community using Gists: developers encouraging others to use Gists when it is first introduced [22], new users commenting on the fact that they are trying Gists for the first time [23] developers or teams setting rules or community procedures around Gists [24][25], and new technical reasons for using Gists (e.g. the mailing list is set to reject attachments [26], or the mailing list software itself rejects messages with code in them [27]). Like with Pastebin.com, there is even occasional shaming when others do not use Gists [28].

Some developers also make the case for Github Gists as an improvement to the mailing list experience in terms of increased "social programming" power [23] as follows:

> I've never used pastebins before. I heard about them the other day when I got roundly booed for pasting some code into an irc window. Can someone tell me how they work?
>
> This lets someone see that file with syntax highlighting in a browser, instead of in an IRC window, etc.(…) I think it's kinda cool to be able to collaborate that way with someone who might not know much about git or source control tools in general, better than posting revisions (…) or sending patches around on mailing lists.

### 4.3 jsFiddle Discussion

jsFiddle represents an interesting case of a language-specific (JavaScript) online IDE which also provides static, sharable URLs. As such, jsFiddle does not have as broad a potential developer user base as Pastebin.com or Gists, both of which are language-neutral. Its adoption curve is therefore smaller than the other two (Figure 3). Also, jsFiddle is designed to help a developer code and test in the browser, so it will not be effective for error logs or other non-code messages. Finally, most of the mailing lists indexed by MarkMail are for FLOSS projects that are written in high-level, multi-purpose, non-browser languages (for example Java, Python, Ruby, C++), so they have little use for a language-specific interface-driven tool like jsFiddle. (We also ran into some issues with the way MarkMail collected data that affected jsFiddle, and these are discussed in the Limitations section.) Another study did note the popularity of jsFiddle on Stack Overflow [29], finding that it was the sixth most common domain linked in postings.

The first mention of jsFiddle on the mailing lists in MarkMail was in January of 2010, but it took a year for jsFiddle mentions to accelerate to more than 100 mentions per month (0.03% of messages). As we mentioned in Section 3.1, we had to clean the data to remove one particular mailing list (jquery-br) because a list-wide email signature line was artificially inflating the numbers for both JS Bin and jsFiddle. However it is worth pointing out that the reason that the list members put that request in their signature lines ("Use JSBIN.COM / JSFIDDLE.NET for code") in the first place was as a way of directing the community towards a preferred behavior. This was the same pattern that we noticed in Pastebin.com, and Github Gists as well: the developers themselves would decide to adopt the innovation and then attempt to encourage others to use it by modeling positive behaviors, social norming and rule-setting, or shaming. Here is an exchange between a new user and a more experienced developer regarding jsFiddle [30]:

> Maybe you can get your issue into jsFiddle?
>
> Thanks for the suggestion to use jsfiddle. It helped to quickly allow me to test out my solution. You can see it in action here...

## 5 Results

Our first research question asks what the rate of diffusion is for pastebins among software developers using mailing lists to communicate. The graph in Figure 2 shows the number of times popular pastebin tools were mentioned in messages to the 8800 mailing lists on MarkMail. The graph shows the classic DoI adoption S-curve: a slow ramp up, followed by a steeper period of increasing usage, which eventually levels out.

Our second question asks whether the diffusion curve differs between different variants of the innovation. Figure 3 shows different shapes to each of the top four pastebin tools. The "first mover" tools either took a long time to ramp up (Pastebin.com) or never quite took off (Pastie). The later-to-market tools were adopted quite quickly and enthusiastically by either a general audience (as with Github Gists) or even in a comparatively narrow niche market (as with jsFiddle).

Our third question is about the stated reasons for and against using this tool to enhance communication on mailing lists. In reading the emails in which developers try to convince their colleagues to adopt the pastebin innovation, they rely on a number of persuasive techniques. First, some developers issue simple entreaties to make everyone's reading experience better. These arguments center on the anti-social aspects of asking for help while simultaneously pasting in thousands of lines of code into a message. We also found pro-pastebin arguments that address the technical downsides to mailing lists, namely that they are overrun with spam and therefore may have been configured to have length limits or rules disallowing attachments.

But we find the most interesting group of arguments are those which claim that a pastebin or online IDE will actually make the programming process more social and therefore more effective. This is especially true in the case of Github Gists (which can be forked, and pull requests issued) and in the case of jsFiddle (which can be edited and the new URL re-sent). In this case, a pastebin is more than a simple link-to-text scheme; it actually adds a layer of social enhancement that did not exist with simple mailing lists.

We also found some anti-pastebin sentiment. A number of developers point out the negative side effects of linking to code rather than archiving it inside the message. Some point out the inconvenience of having to click a link, and many more mention the risk that the link may be broken over time and is therefore unreliable as a record of what has been done in the code and why. However, based on the diffusion curves, this argument does not seem to have much "won" on developer mailing lists as a whole.

## 6 Limitations

**Definition of Diffusion.** Our method of calculating the diffusion rate may differ from the traditional interpretation of "diffusion" because of our use of message counts and not sender counts in our calculation of the adoption rate. In the traditional DoI literature, diffusion is usually defined as "number of people who adopt an invention". In the realm of email, adoption can be described as the number of senders who mention the tool, or it could be the number of times (emails) a sender mentions the tool. Table 3 outlines the differences in calculating diffusion using these two different methods.

**Table 3.** Two different methods of counting diffusion of pastebins on mailing lists

| Method 1 | Method 2 |
|---|---|
| *pm = count of messages mentioning a tool* | *ps = count of senders mentioning a tool* |
| *m = count of all messages* | *s = count of all senders* |
| *diffusion=pm/m* | *diffusion=ps/s* |

In this paper we choose to measure *pm/m* (method 1) rather than *ps/s* (method 2). We determined that it is actually more accurate to count messages rather than senders because for example, a single sender (e.g. "bugzilla@redhat.com") may actually represent an unknown number of other developers on the project. For example, suppose five different developers submit their bugs to a Bugzilla bug-tracking system, and they all used a pastebin for the error log or code sample that they are attaching to the bug tracking email. These five bug reports will all be forwarded to the mailing list as a single sender: bugzilla@company.com. Our method recognizes these messages as representing five instances of pastebin use, rather than just one. We also acknowledge that it is considerably easier to measure messages than to disambiguate senders (who may have multiple email addresses or different aliases).

**Pre-Pastebin Code-via-URL Sharing.** One other limitation of our study is that we do not have a good way for finding out how widespread the practice was of sharing code-via-URL prior to Pastebin.com. The [15] message shows that in November of 2003 there is at least some acknowledgement that it is possible (and may also be desirable) to create a web space for code snippet or log message rather than posting it in the email message. However, in this paper we do not examine how prevalent that practice was. It would be quite challenging to gather this count, since any URL would be a potential code location.

**MarkMail Limitations.** As a source for keyword searching through very high volumes of emails, MarkMail is a good (but not perfect) source. We recognize that MarkMail does not collect every FLOSS mailing list to begin with, and we acknowledge that some of its lists are not about FLOSS. More important for this project, we noticed that MarkMail may occasionally stop collecting from a mailing list abruptly, even though the list is still being used. We had some issues with MarkMail stopping the collection of one small mailing list in June of 2013. This abrupt stoppage affected our graph for jsFiddle since that one, small mailing list did include a very high number of jsFiddle users. (We turned in a support ticket to MarkMail about why they dropped the "jsknockout" mailing list from its collection, but it has so far gone unanswered.) Still, for searching a large number of lists over a long period of time, MarkMail is useful.

## 7 Future Work

We are excited about some avenues of future work, particularly in studying pastebin tools and the innovation curve on two additional sources of FLOSS social programmer communication: IRC and Stack Overflow. These two tools are heavily used by

active social programmers, especially in developing FLOSS and in distributed programming teams of all kinds. They are each very different ways of communicating, and have their own traditions, expectations, and culture. We suspect that the diffusion curves of the pastebin tool will look very different on each of these. Early evidence shows that pastebin tools caught on very quickly on IRC, in some cases producing an even sharper diffusion curve than on mailing lists. In contrast, we have encountered evidence of much more resistance to usage of pastebin tools in certain Stack Overflow postings but not in others. We suspect that this resistance is grounded in fear of link rot, as dead links are considered substantially more serious on a long-term reference site (which Stack Overflow aims to be) than on IRC and mailing lists. We also notice that some sub-communities within Stack Overflow are more accepting of pastebins (and online IDEs in particular) than other sub-communities. We are in the process of analyzing the IRC and Stack Overflow data and comparing it to mailing lists in order to provide evidence for how pastebins diffused on these other communication channels, and why.

## 8 Conclusion

Studying pastebin adoption on developer communication channels, mailing lists in this case, can further our understanding of how innovations diffuse in general, and in particular we can learn about how social media are impacting the traditional communication tools used by software developers. Will developers adopt a "social programming" innovation in order to enhance a tool that has already been in use for nearly forty years? In this paper we were able to confirm that the overall pastebin adoption does follow the S-curve of traditional DoI theory. Within particular tools, we find some differences in rates of diffusion depending on whether the tool being adopted was one of the earlier ones or one of the later ones. Most interestingly, we find that the reasons given for adopting the innovation include both very practical enhancements to the email communication medium, as well as attempts to improve the social aspects of collaborative coding.

## References

1. Apache Software Foundation. Mailing Lists. http://www.apache.org/foundation/mailing-lists.html
2. Apache Software Foundation. Project Management Committee Guide. https://www.apache.org/dev/pmc.html#mailing-list-naming-policy
3. Storey, M.A., Singer, L., Cleary, B., Filho, F.F., Zagalsky, A.: The (R)Evolution of social media in software engineering. In: Proceedings of the on Future of Software Engineering, FOSE 2014, pp. 100–116. ACM (2014)
4. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the 2006 International Workshop on Mining Software Repositories, pp. 137–143. ACM (2006)
5. Pastebin.com. Frequently Asked Questions. http://pastebin.com/faq#1
6. Brian, M.: Pastebin: How a popular code-sharing site became the ultimate hacker hangout. The Next Web, June 5, 2011. http://tnw.to/1CUpN

7. Kelion, L.: Pastebin to hire staff to tackle hackers' 'sensitive' posts. BBC News – Technology, April 1, 2012. http://www.bbc.com/news/technology-17544311
8. Rogers, E.M.: Diffusion of Innovations, 5th edn. Free Press (2003)
9. Lyytinen, K., Damsgard, J.: What's wrong with diffusion of innovation theory? the case of a complex and networked theory. In: Proceedings of the IFIP TC8 WG8.1 Fourth Working Conference on Diffusing Software Products and Process Innovations, pp. 173–190. Kluwer (2001)
10. Chapman, W.B.: PEAR DB_DataObject. On Pear-general list (2003). http://markmail.org/message/elidvojuxpb74kps
11. https://github.com/amberksmith/pastebin_data
12. Rasmussen, K.: Re: RFC: associations with :accessible => true should allow updating. On Rubyonrails-core mailing list, July 23, 2008. http://markmail.org/message/hhnacladvbfmn2ac
13. Internet Archive Wayback Machine Browse History for http://pastebin.com, http://web.archive.org/web/20020901000000*/, http://pastebin.com
14. Internet Archive Wayback Machine. http://web.archive.org
15. Khalid, B.: Re: BTML 2.0 released!!! On PHP-general list, November 6, 2003 http://markmail.org/message/7o7agkvsxwa4khsy
16. Unknown. [Newbie Guide] For the benefit of new members. On PHP-general mailing list, November 5, 2003. http://markmail.org/message/bvg6bqpthu6nhorf
17. Kumar, M.S.: [Newbie Guide] For the benefit of new members. On PHP-general mailing list, April 11, 2004. http://markmail.org/message/xmzgihotajd5ycar (last accessed April 29, 2014)
18. Wells, C.: Overload() problem. On PHP-general list, December 22, 2003. http://markmail.org/message/67vqonuzqf46qfje
19. Stepniewski, L.: Twisted Weekly News #11. On Twisted-Python mailing list, Feburary 17, 2004. http://markmail.org/message/tuqdi3ngd6jbngxi
20. Brown, N.: Re: 2 drives failed. On Linux-Raid-Vger list, January 29, 2010. http://markmail.org/message/lg6ohoycbiadu6j4
21. Rice, K.: Jira and the PasteBin…[Please READ ME]. On Freeswitch-users list. http://markmail.org/message/76gjydtvymg34sfc
22. Brown, G.: Re: Where to upload a ruby script to share it? On Ruby-lang Ruby-talk list, August 4, 2008. http://markmail.org/message/vaai47hafrjl3jgn
23. Stejerean, C.: Re: [Chicago] DVCS Workflows? On Python Chicago list, November 20, 2008. http://markmail.org/message/trj3b2r6e5726bdv
24. Kaiser, F.J.: [BP #4342] Announcement: BP-Testcase Page (html). On Blueprintcss list, January 25, 2011. http://markmail.org/message/d45yagbaj6d7vwmv
25. Leeming, C.: How to use the django-users mailing list properly and doing your homework. On Django-Users list, June 30, 2011. http://mark.mail.org/message/jl5lkq6sp6juw5vh
26. Ryaboy, D.: Re: java.lang.OutOfMemoryError when using TOP udf. On Apache Hadoop Pig User list, November 21, 2011. http://markmail.org/message/4jmb54whfvbenyqy
27. Rowe, S.: Re: Implement price range filter: DataImportHandler started. Not Initialized. No commands can be run. On Lucene Solr-User list, Feburary 14, 2013. http://markmail.org/message/3jguuhkgqdztynht
28. Demeshchuk, D.: Re: riak on one node. On Riak-Users list, October 1, 2012. http://markmail.org/message/mid5slijhzyp7xt3
29. Gomez, C., Cleary, B., Singer, L.: A study of innovation diffusion through link sharing on stack overflow. In: Proceedings of 10th Working Conference on Mining Software Repositories, pp. 81–84. ACM (2013)
30. Rpn. Re: Noob Question: Trouble binding computed value to table header. On Knockoutjs List, September 27, 2012. http://markmail.org/message/m43utkaz4eo5xffa