Evaluating the Quality and Quantity of Data on Open Source Software Projects

Austen Rainer
School of Computer Science
University of Hertfordshire
College Lane Campus
Hatfield
Hertfordshire AL10 9AB
U.K.
a.w.rainer@herts.ac.uk

Stephen Gale
School of Computing Science
Middlesex University
Tottenham Campus
White Hart Lane
London N17 8HR
U.K.
s.gale@mdx.ac.uk

Abstract **– In this paper, we provide a preliminary evaluation of the quality and quantity of data on open source (OS) projects, provided at the SourceForge.net portal. We have derived a dataset of approximately 50000 projects from SourceForge. Using several indicators of project activity, we identify two samples from the entire dataset: the 'most active' OS projects (a total of 456 projects, ~0.9% of the entire dataset), and those projects with active code development (5826 projects, ~11.6%). The number of projects that are active across *all* of our main indicators of activity account for less than 1% of the projects on the portal. This suggests that many OS projects being registered on SourceForge are 'impulse' projects, which do not gather sufficient interest from developers or users to 'activate' those projects and make them 'successful'. It also suggests that researchers, developers and users should be careful about how they use OS portals.**

I. INTRODUCTION

In this paper, we evaluate the quality of data stored for open source (OS) projects on the SourceForge.net portal. We emphasise here that the quality of data is the responsibility of the owner/developers of the respective projects, and the quality of data is not a reflection of the quality of *service* provided by the SourceForge.net portal itself. Evaluating the quality of the data available at SourceForge will help all stakeholder groups (e.g. users, developers, companies and researchers) to make better assessments of the claims made about open source software development.

Longer-term we intend to identify several subsets of the entire dataset. For this paper, we concentrate on comparing two particular subsets: 1) the most active projects in the portal, and 2) those projects that use the portal to only support code development. We relate both of these sub-samples to the entire dataset.

II.BACKGROUND

A. The development of portals for hosting open source projects

Traditionally, OS projects have provided their own online development environments. However, as the resources and infrastructure for coordinating an OS project have stabilised, and dynamic web content technology has matured, OS portals have been created which provide template environments in which to create and host OS projects. Notable examples are SourceForge.net (www.sourceforge.net) and freshmeat.net (www.freshmeat.net). For more information on the typical tools and infrastructure in OS projects, see [1] and [2]. By providing resource and infrastructure, the overhead of creating and supporting a new OS project is reduced. The reduction in overhead brings many advantages. OS portals make it easier for those wishing to initiate a new project to do so, and also enables a new project to be visible from its conception (which in turn will help to attract interested developers and users). OS portals also encourage and support communities of developers and users. For developers, the portals provide a common environment in which projects are aware of each other, and developers (and users) can move freely between projects without having to adapt to a new development environment. For users, the portals provide a gateway to a wide range of applications or code.

The reduction in the overhead of creating and supporting projects also presents certain threats. As projects are now easier to initiate, there is increased likelihood that projects will be created on impulse, resulting in projects that quickly become inactive. With an increasing number of (inactive) projects, many of which are in their early stages of development, it can become increasingly difficult to attract new developers and users. (This can occur if the number of projects is increasing, but the number of developers and users in the community are not increasing at an equivalent rate.) A potential major consequence of this situation is a portal with a vast number of registered projects, but with a very small number of projects that are actually active.

B. The value of portals for supporting research

Researchers across a number of disciplines are increasingly interested in open source software development. Originally, these researchers would turn to the online development environments developed for specific projects to gather data. The popularity of portals hosting OS projects has grown immensely in recent years, with the larger portals now hosting tens of thousands of projects; this has made portals increasingly attractive to researchers. Quantity, however, is not always a good measure of quality. As noted above, an OS portal could be in a situation where it hosts a vast number of inactive projects, including a vast number of projects that have (in a sense) never been active. Just as the number of inactive projects presents problems for developers and users, so the number of inactive projects presents problems for researchers. The researcher needs to identify those possibly small number of relevant OS projects (relevant to the researcher's investigation) amongst a potentially vast number of irrelevant projects.

Perhaps the greatest strength of a dataset of OS projects is its use as a tool for the sophisticated selection of samples of projects. By creating samples where each project is known to possess certain static and/or dynamic properties

it becomes possible to analyse OS in a more controlled and systematic way. Following on from this, the analysis of an OS dataset also enables researchers to be aware of various, perhaps unexpected, properties of the dataset. With SourceForge.net, for example, many projects are not, and have never been, active; and most projects, active or otherwise, are developed by very small numbers of developers – usually one. The creation of such datasets also provide a basis for enabling comparisons between different portals, enabling researchers to assess which portal(s) hosts projects most suited to their studies.

C. The quality of a dataset

In this paper, we refer simply to the quality of the data, or to the quality of the dataset, and we emphasise that quantity of data is not a good indicator of quality of data. We can, however, quickly start to make some distinctions between various 'facets' of quality. For example, two OS projects may use MySQL v4.0. One of these projects could describe itself on SourceForge.net as using MySQL v4.0, whilst the other could describe itself as using MySQL. The first project is being more precise in its description. As a contrasting example, there could be certain aspects of an OS project (e.g. the severity of a bug, or dependencies on other OS projects) for which SourceForge.net does not provide an explicit data field in which to record that aspect. This is an issue of the completeness of description i.e. how completely SourceForge (or indeed any data repository) can describe something. Other facets of quality (e.g. fitness for purpose) could also be considered. We recognise that considerable more thought needs to be directed at how we (and others) should define the quality of an OS dataset. For pragmatic reasons, in this paper we can only recognise this issue, and plan to address it in further research.

III.THE SOURCEFORGE DATASET

A. An overview of the SourceForge.net portal

SourceForge is by far the largest OS portal and claims to currently host over 90,000 projects. (At the time of our data collection the portal claimed to host approximately 85,000 projects.) SourceForge stores a set of common attributes for all projects; these are divided into two groups, the first being static information about the project (such as the license it is released under), and the second containing either derived or statistical information (such as the number of code changes committed to CVS). These attributes are presented by the portal on each project's portal summary page.

B. A summary of the data collection and verification processes

The dataset was collected in a number of stages. During the data collection, we took account of the recommendations given in [3] regarding the perils and pitfalls of automated data collection from portals.

The first task was to build a list of available projects. As the portal in question did not provide a ready-made list, we needed to create our own. We considered using the 'Software Map' provided by the portal, and also the activity-ranking pages. Both presented problems e.g. many projects have not positioned themselves in the 'Map' and the activity-ranking excludes those projects with 0% activity. Finally, we decided to use a Perl-based web-crawler to search for projects with any common three-letter character sequence in their project description (the minimum allowed by the facility). We derived a list of approximately 70,000 projects. Once a list of projects was obtained, a Perl script was used to download the textual content of each project's information page. Projects for which no information page could be retrieved were discarded. A further consideration of the downloaded pages revealed a problem with some of the statistics reported by SourceForge.net, further reducing the usable dataset to approximately 50,000 projects.

In order to verify that the data had been parsed correctly, fifty projects were chosen at random from the list, then the set of extracted data fields belonging to those projects were compared to the original, online project pages. Three sets of comparisons were made:

- Checking that the extracted values for every field were correct.
- Checking that any missing fields were also missing on the original page.
- Checking that the structure of the output had remained consistent between projects.

This testing uncovered a number of flaws in the data extraction process, mostly caused by idiosyncrasies in the formatting of the pages. Other errors came from unexpected attributes of some fields, for example, the legality of a project reporting several concurrent development statuses, or reporting the use of the same programming language twice. Where possible discrepancies were corrected, otherwise we dropped the project from our dataset.

The final output of this process was a tab-delimited file, with columns for each identified attribute, and one project on each row. The details of specific fields are given in a later section. We developed two versions of the dataset: a simpler version (consisting of only those attributes that contained single values) for analysis using SPSS, and a more complex version (which includes those attributes with multiple concurrent values) for analysis using MySQL.

Since carrying out this data collection, we have discovered that the OSSmole project (e.g. [4]) has released a large dataset of the projects on SourceForge.net. This project is also hosted on the SourceForge.net site (at http://ossmole.sourceforge.net/)

C. An overview of the dataset used in our evaluation

A summary of the information we have collected is presented in Table 1. The table indicates that several attributes could contain multiple concurrent values. For example, a project could be developing a software system using more than one programming language. Multiple concurrent values make it difficult to analyse a dataset, hence multi-valued attributes were expanded to give a set of binary properties, or "flags". This resulted in a total of 424 properties for each project.

We also make a distinction between those attributes (properties) that can be used to represent project activity, and those attributes (properties) that can be used to describe the characteristics of the projects. Longer-term,

we want to investigate the relationship between project activity and project characteristics. For this paper, we concentrate only on project activity.

Given the number of projects, and the number of properties for each project, this is clearly a very large software engineering data set. We are aware of Healy and Schussman's [5] study of 46,356 OS projects, based on a SourceForge dataset provided to them in August 2002. And, as previously noted, we are aware of the OSSmole project which also provides information on a large number of SourceForge.net projects.

There are several potential problems with datasets of such size: that the size of the dataset is not an indication of the dataset's quality; that such a large dataset could have a considerable degree of diversity in it; that such a large dataset is extremely difficult to verify for accuracy; that datasets of this size need some preliminary re-organisation (which can require considerable time and effort and could introduce its own errors); and that such a dataset provides 'snapshot' data on the overall status of the projects at one point in time, and does not show the changes that have occurred over time within each project.

Table 1 Summary of data collected for each project

Category of attribute	**Attribute**	**Number of concurrent values**
	Project name	1
	Registration Date of project	1
Project activity (Major indicator)	Number of Commits	1
	Number of files added to CVS	1
	Number of Developers	1
	Number of Forum Messages	1
	Number of Forums	1
	Number of Mailing Lists	1
	Total number of bugs	1
	Total number of technical support requests	1
	Total number of patches	1
	Total number of feature requests	1
Project activity (Minor indicator)	Number of open Bugs	1
	Number of open technical support requests	1
	Number of open patches	1
	Number of open feature requests	1
Project characteristics	Development status	7
	Environment	12
	Intended audience	14
	License	57
	Operating system	30
	Programming language	42
	Topic	185
	Natural language	60
	Has released files	1
	Total number of properties	424

IV. A SUMMARY OF THE PROJECTS IN THE ENTIRE DATASET

Table 5 provides a summary of the distribution of values for the project-activity properties of all the projects in the entire dataset. The table provides some interesting insights:

1. The modal value for *all* of the properties is the value assigned, by default, by the portal when the project is first created For example, at least one developer must be registered with a project, and the web portal automatically produces two forum messages and, presumably, two forums[1].
2. The median value for all of the properties is also the default value. This indicates that for each attribute at least half of the projects in the web portal are 'empty'!
3. The mean, mode and median averages for number of developers supports Krishnamurthy's finding [6] that most projects have only one or two developers. Our findings are on a considerably larger scale that Krishnamurthy's.
4. Some of the maximum values are surprisingly high when one considers the typical values. For example, there is at least one project with 262 developers, a project with over 30,000 forum messages, a project with almost 140,000 commits, and a project with over 26,000 files added.
5. There are some suggestions for different sub-sets of data. These are summarised in Table 2.

V. THE MOST BROADLY ACTIVE PROJECTS

Table 3 summarises the indicators of project activity, and identifies thresholds that can act as selection criteria for selecting a sub-sample.

The properties 'Number of developers', 'Number of forum messages' and 'Number of forums' are special cases. When a project is registered with the web portal, the portal automatically sends two forum messages. This sending of the messages also implies that the portal also automatically creates a forum. And there must be a developer who owns the project on the portal.

For our first sub-sample, we identified those projects that are active in all of the activity indicators. Phrased another way, the sub-sample consists of those projects that meet or exceed the thresholds defined in Table 3. Our sub-sample consists of 456 projects, ~0.9% of the entire dataset of 50012 projects. While the sub-sample is very small compared to the entire dataset, such a sample is still large enough to permit substantive investigation. (By way of comparison, there are few, if any, datasets used in software estimation that are of a size similar to this sub-sample.)

Table 6 provides a summary of the distribution of values for the sub-sample we have selected. It is clear that this sub-sample is much 'richer' than the entire dataset e.g. note the distribution of values across the percentile breakdowns.

A comparison of Table 6 with Table 5 reveals that our sub-sample does not include all projects with the maximum values for properties. For example, in this sub-sample (Table 6) the largest number of developers on a project is 132, whereas for the entire dataset (Table 5) the largest number of developers on a project is 262.

Notice that the typical values (mean, median and mode) in Table 6 are small relative to the maximum values. This clearly suggests something particularly unusual about the projects with the maximum values. For example, the sub-sample contains at least one project with 965 patches.

VI. PROJECTS USING THE PORTAL FOR CODE DEVELOPMENT ONLY

We speculate that there is a sub-sample of projects on SourceForge where developers are using the portal to only *develop* code, in contrast to making feature requests, reporting bugs, posting patches etc.

Using a revised set of selection criteria (see Table 4) we selected a second sub-sample from the entire dataset, this sub-sample consisting of 5826 projects. Of the 456 projects in sub-sample 1 (the most active projects), 397 (87%) of those projects are also in sub-sample 2 (code development only). This indicates that these samples are not mutually exclusive. Our phrase 'code development only' is probably slightly misleading: it refers to those projects where there is *at least* code development activity. For many projects, code development activity is the only activity.

The distribution of values for this sub-sample is shown in Table 7. Again, it is clear that this sub-sample (like the most-active projects sub-sample) is much 'richer' that the entire dataset. The code-development-only sub-sample appears to have less 'breadth' of activity than the most-active-projects sub-sample (compare the averages and the range of values across all of the properties for the two sub-samples).

VII. DISCUSSION AND CONCLUSIONS

A. Summary of our findings

The analysis we report here was motivated by the awareness that although OS portals can contain a vast number of OS projects, the raw number of projects is not a good indication of the quality of data being 'stored' for those projects.

Our analysis shows that many of the projects currently hosted on the SourceForge.net portal are not, and have never been, active on the portal. The number of projects that are active across *all* of our main indicators of activity account for less than 1% of the projects on the portal. Using less stringent selection criteria (i.e. selection based only on activity indicators that suggest code development) provides a larger sub-sample comprising approximately 11.6% of the entire dataset.

[1] While the portal automatically creates two forums it seems that many project administrators delete one of the forums.

Table 2 Sub-sets of the entire data

Sub-set	Properties
The most broadly active projects	All of the main activity indicators have non-default values.
Coding-active but not user-active	The *Number of commits*, *Number of file adds*, *Number of developers*, and Low user community activity attributes have non-default values
User-active but not coding-active	A sample defined by the thresholds: *Number of developers* < 3, *Number of commits* is low, *Number of file adds* is low, *Number of forum messages* is high, *Number of forums* is high, *Mailing list* is high, *Number of feature requests* is high
'Good intention'	Low coding activity and low user activity.

Table 3 Indicators of activity

Property	Thresholds for activity
Number of Commits	> 0
Number of Adds (files added to CVS)	> 0
Number of Developers	> 0
Number of forum Messages	> 2
Number of Forums	> 1
Number of Mailing Lists	> 0
Total number of Bugs	> 0
Total number of Tech support requests	> 0
Total number of Patches	> 0
Total number of Feature requests	> 0

Table 4 Selection criteria for code development-only projects

Property	Thresholds for activity
Number of Commits	> 2
Number of Adds (files added to CVS)	> 0
Number of Developers	> 1
Number of Mailing Lists	> 1

Table 5 Distribution of values for the project-activity properties, for all projects in the entire dataset (n=50012)

				Dataset percentiles							
Attribute	**Mean**	**Mode**	**Median**	**05**	**25**	**50**	**75**	**95**	**99**	**Min**	**Max**
Number of commits	173	0	0	0	0	0	28	711	3137	0	138928
Number of file adds	58	0	0	0	0	0	11	241	997	0	26008
Number of developers	2	1	1	1	1	1	2	7	16	0	262
Number of forum messages	24	2	2	2	2	2	3	25	340	0	30357
Number of forums	2	2	2	1	2	2	2	3	4	0	28
Number of mailing lists	1	0	0	0	0	0	1	3	5	0	44
Total number of bugs	6	0	0	0	0	0	0	14	105	0	8131
Total number of tech requests	2	0	0	0	0	0	0	1	9	0	73342
Total number of patches	1	0	0	0	0	0	0	1	10	0	2896
Total number of feature requests	2	0	0	0	0	0	0	6	38	0	2559

Table 6 Summary data for the most broadly active projects (n=456)

				Percentile breakdown							
Attribute	**Mean**	**Mode**	**Median**	**05**	**25**	**50**	**75**	**95**	**99**	**Min**	**Max**
Number of commits	2054	66	669	42	230	669	1809	6509	25091	1	138928
Number of file adds	479	13	134	5	42	134	417	1617	7758	1	20767
Number of developers	9	2	5	1	2	5	10	29	73	1	132
Number of forum messages	455	6	90	5	30	90	364	2002	5751	3	13790
Number of forums	3	2	3	2	2	3	3	5	9	2	28
Number of mailing lists	3	1	2	1	1	2	3	5	12	1	17
Total number of bugs	103	24	34	5	18	34	94	397	1254	2	2307
Total number of tech requests	20	1	5	1	2	5	13	48	307	1	1942
Total number of patches	15	1	5	1	2	5	13	43	209	1	965
Total number of feature requests	39	1	12	1	4	12	34	166	488	1	1275

Table 7 Distribution of values for code-development-only projects (n=5826)

Attribute	Mean	Mode	Median	Percentile breakdowns						Min	Max
				05	25	50	75	95	99		
Number of commits	959	3	236	9	65	236	818	3728	10057	3	138928
Number of file adds	310	1	78	2	20	78	251	1180	3716	1	26008
Number of developers	6	2	4	2	2	4	7	18	36	2	262
Number of forum messages	130	2	4	0	2	4	19	392	2272	0	30357
Number of forums	2	2	2	1	2	2	3	4	6	0	28
Number of mailing lists	2	1	2	1	1	2	3	5	8	1	44
Total number of bugs	33	0	2	0	0	2	14	134	541	0	8131
Total number of tech requests	4	0	0	0	0	0	1	11	62	0	1942
Total number of patches	5	0	0	0	0	0	1	15	81	0	1964
Total number of feature requests	12	0	0	0	0	0	5	51	197	0	2353

B. Conclusions

Based on our analysis, our conclusions are:

1. That developers, users and researchers should be careful about how they use open source portals. This is not to say that open source portals (and indeed open source projects and products) are not valuable; rather, that users of these portals need to be careful about how they use these portals so that they can gain maximum benefit from the portals.
2. That the research community should conduct an evaluation of the quality and quantity of data available at these portals prior to commencing any more specific analyses. Such an evaluation can strengthen confidence in subsequent analyses performed by researchers, including increasing confidence in the generalisability of any claims that can be made from such subsequent analysis. For example, given that only 0.9% of the projects are classified as most-active, it is unlikely that a researcher would *randomly* select one of these most-active projects from the SourceForge.net portal. (Compare this with the typical threshold of 5% for Type I errors set by statisticians.) It is possible of course that a researcher has some prior knowledge of a more interesting project, but the problem still remains: to what degree does this 'interesting project' represent all projects within the portal, or indeed a certain class of projects. (This is setting aside the issue that choosing an *a priori* interesting project potentially introduces known biases into the analysis.)

C. Further research

In further research, we intend to clarify our selection criteria for identifying sub-samples, identify some further sub-samples, compare these sub-samples based on the activity indicators, and then compare the sub-samples based on the other attributes summarised in Table 1. We also want to consider, in much more depth, the concept of a 'quality dataset' of OS projects.

We are also conscious of the load that web-crawling places on the servers of OS portals. Consequently, we want to work with other researchers to encourage the sharing of existing datasets in order to reduce that load.

ACKNOWLEDGEMENTS

We should like to thank the reviewers for their helpful comments. We also thank SourceForge.net for the services it provides to developers and users and (indirectly) to researchers.

REFERENCES

[1] J. E. Robbins, "Adopting OSS Methods by Adopting OSS Tools," presented at 2nd Workshop on Open Source Software Engineering, 24th International Conference on Software Engineering, Orlando, Florida, 2002.

[2] J. Holck and N. Jorgensen, "Do not check in on red: Control meets anarchy in two open source projects," in *Free/Open Source Software Development*, K. Stefan, Ed. Hershey (PA): Idea Group Publishing,, 2004, pp. 1-26.

[3] J. Howison and K. Crowston, "The perils and pitfalls of mining SourceForge.," presented at Mining Software Repositories Workshop, International Conference on Software Enginnering (ICSE 2004), Edinburgh, Scotland, May 25., 2004.

[4] M. S. Conklin, J. Howison, and K. Crowston, "Collaboration using OSSmole: a repository of FLOSS data and analyses," presented at Mining Software Repositories (MSR) Workshop at International Conference on Software Engineering, May 17 2005, St Louis, Missouri, 2005.

[5] K. Healy and A. Schussman, "The ecology of F/OSS software development," 2003.

[6] S. Krishnamurthy, "Cave or community? an empirical investigation of 100 mature Open Source projects," *First Monday*, vol. 6, 2002.