Sampling open source projects from portals: some preliminary investigations

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 50 000 open source (OS) projects hosted at the SourceForge.net portal. Using several indicators of project activity, we identify one sample from the entire dataset: the 'most-broadly-active' OS projects. 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. 75% of the projects currently hosted on the SourceForge.net portal are not, and have never really been, active on the portal. Furthermore, whilst there has been a substantial increase in the number of projects being added to SourceForge.net over time, the number of projects being added that then go on to become most-broadly-active projects seems to be decreasing over time. Finally, we recognise that care needs to be taken in defining samples, such as the most-broadly-active projects, as these definitions raise implications for the conclusions that one makes and the generalisations that one should draw.*

1. 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 the entire dataset with one sample from that dataset: the most-broadly-active projects.

The remainder of this paper is organised as follows. Section 2 discusses the value of web portals for hosting open source projects. Section 3 describes our methods for collecting and organising our dataset. Section 4 provides a summary of the entire dataset. Section 5 provides a summary of the most-broadly-active sample. Section 6 compares the growth of projects for the entire dataset and the most-broadly-active sample. Section 7 briefly discusses our findings, including some caveats. Finally, section 8 provides some brief conclusions.

2. Background

2.1. 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

11th IEEE International Software Metrics Symposium (METRICS 2005)
1530-1435/05 $20.00 © 2005 **IEEE**

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.

2.2. 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.

One of the major advantages of a large dataset of OS projects, for researchers, is that the datasets can support the sophisticated selection of sub-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, the majority of projects are not, and have never really been, active; and most projects, active or otherwise, are developed by very small numbers of developers – usually one. The creation of such datasets and sub-samples also provide a basis for enabling comparisons between different projects hosted on different portals, enabling researchers to assess which portal(s) hosts projects most suited to their studies.

2.3. The quality of the 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 considerablly 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.

3. The SourceForge dataset

3.1. An overview of the SourceForge.net portal

SourceForge is by far the largest OS portal and claims to host almost 100,000 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.

3.2. 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. Notice that this is considerably smaller than the 85,000 projects claimed to be on SourceForge.net at the time we downloaded the project descriptions. Some of the difference between the 85,000 projects and the 70,000 projects could be explained by the fact that some OS projects have not entered *any* description for their project. This could be consistent with OS projects that are created on impulse. As we are not able to easily determine how many of the 15,000 projects have not entered a description, we cannot really estimate the degree to which this lack of description accounts for the large discrepancy between the projects on SourceForge and the projects we have identified.

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. This reduced the number of projects to approximately 69,000. During our analysis we then found some problems with the reporting of data on the SourceForge.net website. Removing projects affected by this problem reduced our data 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, and the set of extracted data fields belonging to those projects were compared to the original, online project pages. Three sets of checks were made:

- Checking that the extracted values for every field were correct.
- Checking that any missing fields in our dataset were also missing on the original page.
- Checking that the structure of the output remained consistent across 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 the next section.) We then 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.

3.3. An overview of the dataset used in our evaluation

A summary of the information we have collected is presented in Table 1. We make a distinction between those attributes that can be used to represent project *activity*, and those attributes that can be used to describe the characteristics of the projects.

The table indicates that almost all of the project characteristics 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".

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 attributes	424

Table 2. Possible samples of the entire dataset

Sample	Definition of sample
The most-broadly-active projects	*All* of the main activity indicators have non-default values for the project. See section 5 for more detail.
Coding-active but not user-active	Values for the *Number of commits*, *Number of file adds*, and *Number of developers* are high, and values for other attributes are low.
User-active but not coder-active	The inverse of the coding-active sample.
‘Good intention’	Low coding activity and low user activity.

11th IEEE International Software Metrics Symposium (METRICS 2005)
1530-1435/05 $20.00 © 2005 IEEE

In other words, rather than having only one multi-valued attribute for programming language, we constructed 42 binary-valued attributes, each attribute relating to one programming language (e.g. the first attribute might indicate the use of Java, the second attribute the use of C++ etc.). The expansion of the multi-valued attributes resulted in a total of 424 overall ‘properties’ for each project.

Most of the major indicators of activity report cumulative values for the duration of the project. The one exception is *Number of developers*, which reports the number of developers currently registered with the project.

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. There are a number of previous studies of OS datasets from SourceForge. For example Healy and Schussman [4] report on a study of 46,356 OS projects, based on a SourceForge dataset provided to them in August 2002. In their analysis, they looked at the entire dataset only and did not identify sub-samples within their dataset. The OSSmole project (hosted on the SourceForge portal itself, at http://ossmole.sourceforge.net/) provides analysis of OS projects at SourceForge.net and, more recently, Notre Dame University have begun to provide datasets of OS projects hosted at SourceForge.net (http://www.nd.edu/~oss/).

There are several potential problems with such large dataset: 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 quality; 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.

4. A summary of the projects in the entire dataset

Table 3 provides a summary of the distribution of values for the major project-activity attributes of all the projects in the entire dataset. The table provides some interesting insights:

1. The modal value for *all* of the attributes 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 attributes is also the default value. This indicates that, for each attribute, at least half of the projects in the web portal are ‘empty’ for that attribute.

3. The percentile breakdowns indicate that for each of the attributes, 75% of the dataset has the default value.

4. The mean, mode and median averages for number of developers supports Krishnamurthy’s finding [5] that most projects have only one or two developers. Our analysis is based on a considerably larger sample than Krishnamurthy’s study.

5. 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, a project with over 26,000 files added, and a project with 73,000 technical requests. (These maximum values are not all be taken from the same project).

6. There are some suggestions for different samples of data. These are summarised in Table 2.

5. The most broadly active projects

Table 5 summarises the major indicators of project activity, and identifies thresholds that can act as selection criteria for selecting a sub-sample.

The thresholds given in the table are conservative, being the minimum non-default values possible for each indicator.

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

Table 3 Distribution of values for the project-activity properties, for all projects in the entire dataset (n=50012)

				Percentiles breakdowns						
	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 4 Summary data for the most broadly active projects (n=456)

				Percentile breakdown							
	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

11th IEEE International Software Metrics Symposium (METRICS 2005)
1530-1435/05 $20.00 © 2005 **IEEE**

Table 5. Indicators of activity and threshold values

Indicator of project activity	Thresholds
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

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 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 5. 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 datasets used in software estimation that are of a size similar to this sub-sample.)

Table 4 provides a summary of the distribution of values for the sub-sample we have selected. The sample is of course not now representative of the projects hosted at SourceForge, but the sample is now smaller, more manageable and more focused. Consequently, the sample should consist of a more suitable subset of data to aid particular kinds of investigation. And by having a better defined sample, one should be able to make more confident generalisations to a population based on that sample.

A comparison of Table 4 with Table 3 reveals that our sub-sample does not include all projects with the maximum values for properties. For example, in this sub-sample (Table 4) the largest number of developers on a project is 132, whereas for the entire dataset (Table 3) the largest number of developers on a project is 262.

6. The growth of projects on SourceForge

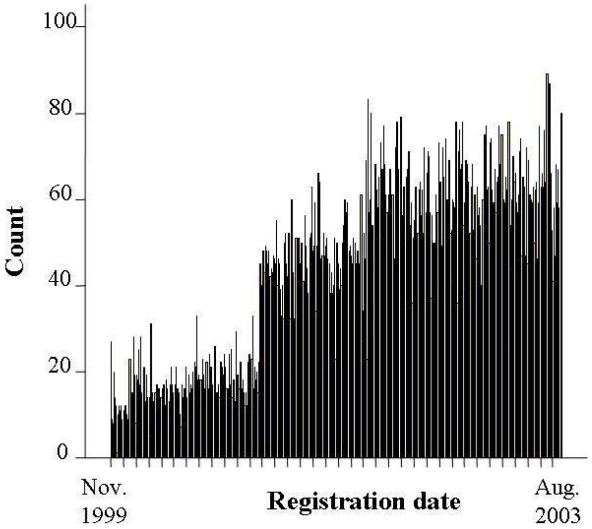

Figure 1. Projects added per day (entire dataset, 1999 - 2003)

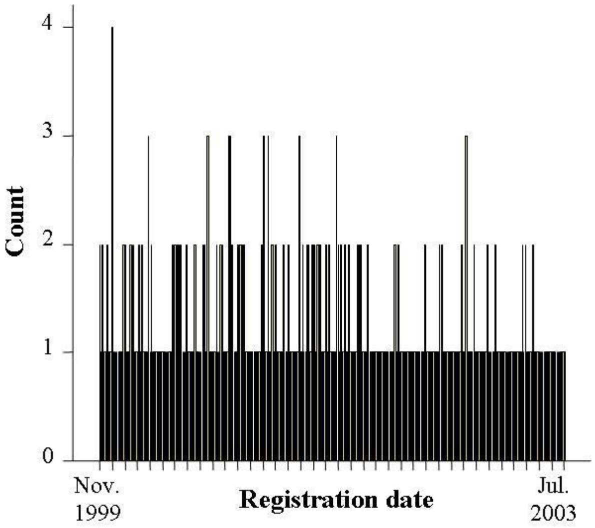

Figure 2. Projects added per day that subsequently became most-broadly-active (1999 - 2003)

Figure 1 presents a bar-chart of the number of projects added to SourceForge.net per day between the end of 1999 and mid-2003. There is a noticeable increase in projects being added to SourceForge.net in early February 2001. Figure 2 presents a bar-chart of the number of most-broadly-active projects added to SourceForge.net over the same period. Comparing the two figures, it is clear that there is no obvious equivalent increase in the number of most-broadly-active projects around early February 2001.

In relative terms, the number of most-broadly-active projects is actually very small (less than 1% of the entire dataset) but the *period of time* over which the most-broadly-active projects are created is broadly the same as all of the projects (approximately 1400 days). Consequently, there may be an equivalent increase in most-broadly-active projects but this increase is 'hidden' by the long period of time over which the small sample is spread. A different method by which we can investigate whether there was an increase in most-broadly-active projects is to examine the average number of projects being created per time period. If there was an increase in the number of most-broadly-active projects then there should be an increase in the average number of most-broadly-active projects being created after February 2001. Stated explicitly, our hypothesis is:

H_1: There is no equivalent increase in the number of most-broadly-active projects (compared to the entire set of projects) after February 2001.

In order to investigate this hypothesis, we distinguished between two periods of time: Phase 1 (November 1999 – January 2001) and Phase 2 (February 2001 – July 2003). These periods of time can be measured in two ways:

1. As the difference, in *days*, between the first and last dates of the time period.

2. As a count of the number of actual dates on which projects were actually created. Because the number of most-broadly-active projects is so small there is more likely to be 'empty' dates for the most-broadly-active dataset.

Overall, we consider that the second method of measuring the time periods leads to fairer averages, however for completeness we report averages using both measures of time period.

Table 6 presents the averages for the entire dataset. Table 7 presents the averages for the most-broadly-active dataset. Table 6 indicates that averages for Phase 2 of the entire dataset are three and half times the averages for Phase 1. These averages are consistent with Figure 1: both the table and the figure show that the average number of projects being added to SourceForge has substantially increased. Note also that the ratios in Table 6 for the two phases are close to 1. This indicates that there are very few 'empty' days in both periods. In other words, projects have been added for almost every day across Phase 1 and Phase 2.

Table 6 Summary statistics for entire dataset (1999 - 2003)

	Count of dates	**Count of projects**	**Average**	**Total duration (days)**	**Count of projects**	**Average**	**Ratio dates / days**
Phase 1	459	6070	13.2	461	6070	13.2	~1.0
Phase 2	935	43942	47.0	935	43942	47.0	1.0
Overall	1395	50012	35.9	1397	50012	35.8	~1.0

Table 7 Summary statistics for the most-broadly-active sample (1999 - 2003)

	Count of dates	**Count of projects**	**Average**	**Total duration (days)**	**Count of projects**	**Average**	**Ratio dates / days**
Phase 1	143	189	1.3	457	189	0.4	0.3
Phase 2	228	267	1.2	888	267	0.3	0.3
Overall	371	456	2.49	1345	456	0.4	0.3

Table 7 presents a very different picture: the averages for Phase 2 of the most-broadly-active dataset are *lower* than the averages for Phase 1. These averages support the hypothesis that there is no equivalent increase in the number of most-broadly-active projects after February 2001. Furthermore, the averages suggest that there might actually be a *decrease* of between 12% and 25% (the approximate percentage difference between the averages 1.3 and 1.2, and the averages 0.4 and 0.3). In other words, although substantially more projects are being added to SourceForge after February 2001 there may actually be *less* projects 'becoming' most-broadly-active. Note also that the ratios in Table 7 for the two phases are much lower than 1. This indicates that there are many days (about three in four days) when there is no most-broadly-active project added to SourceForge (more precisely, there is no project added that subsequently becomes a most-broadly-active project). (Incidentally, the low ratios support our preference for using our second definition of time periods for Phase 1 and Phase 2.)

7. Discussion

7.1. 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 the number of projects that are active across *all* of our major indicators of activity account for less than 1% of the projects on the portal. Further analysis suggests that the number of most-broadly-active projects added to SourceForge appears to be decreasing over time, even though the total number of projects being added to SourceForge is actually increasingly substantially.

7.2. Defining samples and populations

The selection criteria presented in Table 5 could be used as the basis for a *definition* of a population of OS projects. Such a definition can potentially provide a number of advantages to the research community. For example, the definition could:

- Provide a framework with which to conduct literature reviews
- Provide a framework with which to conduct systematic meta-analyses of previous studies
- Provide a framework for replicating previous studies
- Provide a framework for the systematic selection of one or more OS projects for detailed case study
- Support the generalization of findings from one or more case studies
- Support the comparison and consolidation of samples that have been drawn from different OS portals
- Provide a framework by which findings in-the-large (i.e. based on the survey of a large sample of OS projects) can be related to findings in-the-small (i.e. based on a detailed study of a small number of OS projects)

The availability of a definition allows researchers both independence in how they derive and use their own (or others) datasets, as well as a mechanism by which independently-derived datasets can subsequently be 'consolidated'.

7.3. Some caveats

7.3.1. A snapshot view of the projects. Both Table 3 and Table 4 provide a summary of the *overall* status of the projects, and not the current status (or indeed the status at any particular point in time). As noted, most of major indicators of activity report cumulative values for the duration of the project, with the one exception of *Number of developers*, which reports the number of developers currently registered with the project. In order to properly investigate the *current* status of the projects, we would need to collect additional data from the portal. There is some data available within the current dataset (i.e. the minor indicators of activity) that can indicate the current status of the projects.

7.3.2. Open source projects. Not all projects registered on SourceForge are necessarily intended to be about the initial or continued development of some piece of software. Some projects created on SourceForge seem to be about SourceForge providing an opportunity for a developer to host a set of code that others can then use in their own work in other projects. In other words some developers may be using SourceForge as a mechanism for storing and distributing code, rather than as environment within which to collaborate.

7.3.3. The most-broadly-active projects. We have defined our most-broadly-active sample as containing those projects that are active across all of our major indicators of project activity. This definition needs to be treated with some caution. There may be projects that are very active but in only specific areas (as suggested with Table 2). Related to this, there may be projects that are active across all of the areas of activity but choose to not report this data on SourceForge.net (perhaps using another web site to host some of the activity).

An alternative definition of our most-broadly-active sample is that it contains those projects on SourceForge that are using the full range of facilities provided by SourceForge. This implies that for these projects SourceForge is the primary (and perhaps only) Internet 'location' for supporting the activity of the project. While this is a different definition, it still provides a broadly similar implication i.e. we are identifying a 'rich' sample for further analysis. The alternative definition is likely to present different implications for the generalisations of any conclusions we draw from subsequent analyses.

7.4. Lessons learned

Give the size and nature of the dataset, collecting, re-organising and analysing our data from SourceForge.net has consumed a considerably amount of time and effort. It also involved several iterations of data re-organisation and analysis, as we inevitably found errors in our work. We have found it helpful to duplicate our data re-organisation and analysis in two ways: by using two software systems (MySQL and SPSS) to duplicate much of the analyses, and by the two authors independently conducting analysis and confirming the findings.

7.5. Further research

In further research, we intend to: clarify our selection criteria for identifying further sub-samples (particularly the code-active sub-sample); identify and compare sub-samples; consider alternative definitions of the samples (e.g. the most-broadly-active sample vs. those projects that make most use of a portal); explore in much more depth the concept of a 'quality dataset' of OS projects; and investigate if and then how OS projects 'evolve' into the most-broadly-active projects.

8. Conclusions

We have conducted some preliminary analysis of the projects on SourceForge.net in order to identify the quality and quantity of data available for these projects. Overall, we have found that the majority of projects on SourceForge.net are 'empty'. We identified a more focused and 'richer' sample: the most-broadly-active projects. The sample comprises less than 1% of the projects on SourceForge.net. We recognised that care needs to be taking in defining our sample, as the definitions of the sample will have implications for the conclusions and generalisations that we can make. We also found that while there has been a substantial increase in the number of projects being added to SourceForge.net over time, there has been no equivalent increase in the number of most-broadly-active projects, and in fact there appears to be a decrease in such projects over time. We have also suggested that the indicators of activity and their associated thresholds could be used as the basis of a definition of a population of OS projects. Such a definition provides advantages to researchers e.g. supporting systematic sampling, the comparison of samples, and replication.

9. Acknowledgements

We thank the reviewers for their helpful comments on a draft of this paper. We also acknowledge SourceForge for hosting such a large set of open source projects and, as a consequence, indirectly providing us with access to a large dataset.

10. References

[1] J. E. Robbins, "Adopting OSS methods by adopting OSS tools," presented at 2^{nd} *Workshop on Open Source Software Engineering* (co-located with 24^{th} *International Conference on Software Engineering*), Orlando, Florida, 2002.

[2] J. Holck and N. Jorgensen, "Do no check in on red: control meets anarchy in two open source projects," in *Free/Open Source Software Development*, S. Koch, 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 Repositories Workshop* (co-located with 26^{th} *International Conference on Software Engineering* (ICSE)), Edinburgh, Scotland, 2004.

[4] K. Healy and A. Schussman, "The ecology of open source software development," Department of Sociology, University of Arizona. 2003.

[5] S. Kirshnamurthy, "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects," *First Monday*, vol. 7, 2002.