Replicating MSR: A study of the potential replicability of papers published in the Mining Software Repositories Proceedings

Gregorio Robles
GSyC/LibreSoft
Universidad Rey Juan Carlos
Madrid, Spain
Email: grex@gsyc.urjc.es

***Abstract*—This paper is the result of reviewing all papers published in the proceedings of the former International Workshop on Mining Software Repositories (MSR) (2004-2006) and now Working Conference on MSR (2007-2009). We have analyzed the papers that contained any experimental analysis of software projects for their potentiality of being replicated. In this regard, three main issues have been addressed: i) the public availability of the data used as case study, ii) the public availability of the processed dataset used by researchers and iii) the public availability of the tools and scripts. A total number of 171 papers have been analyzed from the six workshops/working conferences up to date. Results show that MSR authors use in general publicly available data sources, mainly from free software repositories, but that the amount of publicly available processed datasets is very low. Regarding tools and scripts, for a majority of papers we have not been able to find any tool, even for papers where the authors explicitly state that they have built one. Lessons learned from the experience of reviewing the whole MSR literature and some potential solutions to lower the barriers of replicability are finally presented and discussed.**

***Keywords*-replication, tools, public datasets, mining software repositories**

Replication package: http://gsyc.urjc.es/~grex/msr2010.

I. Introduction

Mining software repositories (MSR) has become a fundamental area of research for the Software Engineering community, and of vital importance in the case of empirical studies. Software repositories contain a large amount of valuable information that includes source control systems storing all the history of the source code, defect tracking systems that host defects, enhancements and other issues, and other communication means such as mailing lists or forums. As a result of the possibilities that mining software repositories offer, an annual workshop first, then working conference on this topic has been organized with an extraordinary success in participation and research output.

Being mainly focused on empirical research, we wanted to evaluate how much of the research presented at the MSR can be potentially replicated. Replication is a fundamental task in empirical sciences and one of the main threats to validity that empirical software engineering may suffer [1]. Among these threats, we may encounter: lack of independent validation of the presented results; changes in practices, tools or methodologies; or generalization of knowledge although a limited amount of case studies have been performed.

A simple taxonomy of replication studies provides us with two main groups: exact replications and conceptual replications. The former ones are those in “which the procedures of an experiment are followed as closely as possible to determine whether the same results can be obtained”, while the latter ones are those “one in which the same research question or hypothesis is evaluated by using a different experimental procedure, i.e. many or all of the variables described above are changed.” [2]. In this paper, we will target exact replications as the requirements that have to be met to perform an exact replication are more severe, and in general make a conceptual replication feasible.

We are focusing in this paper on potential replication as we have actually not replicated any of the studies presented in the papers under review. Our aim in this sense is more humble: we want to check if the necessary conditions that make a replication possible are met.

The rest of the paper is structured as follows: in the next section, the method used for this study is presented. Then some general remarks on the MSR conference are given, to give the reader a sense of the type of papers that are published in the MSR proceedings. Results will be presented in section IV: first, the replication-friendliness of the papers will be shown and then each of the individual characteristics that we have defined will be studied independently. MSR has a special track called the “Mining Challenge”, a section is devoted to analyze it with the aim of finding if results differ from those for the rest of papers. Then, other non-quantitative facts from the review are enumerated. Section VII discusses the findings of the paper and hints at possible solutions. Then, conclusions are drawn. In a final section, the replicability of this paper is considered.

II. Method

The method that has been used to perform this study is a complete literature review of the papers published in the proceedings of any of the MSR workshops/working conferences up to date.

For each paper, we determine its nature in order to proceed with a subsequent analysis. So, although the MSR is in general a conference with empirical papers that propose new methods and perform case studies, some papers are non-experimental. For this type of papers, replication cannot be achieved, so we have not proceeded to gather data on them.

On the other hand, a subgroup of the papers that are of special interest for this research on replicability are the ones from the "MSR Challenge", an opportunity to researchers to apply, compare and challenge methods and tools with a common repository. The MSR Challenge has taken place since the year 2006 and accepted "challenge" papers have been in general published in the proceedings as short papers.

Figure 1 shows a diagram with the general flow that has to be accomplished for the study of publicly available data sources found on the Internet [3]. The process starts with the identification and localization of the data source, if followed by its retrieval to a local machine, a data extraction procedure that may include parsing, cleaning, filtering and possibly other data treatment, and its storage into a convenient form (usually a database) to be analyzed. Depending on the source, the amount and the completeness of the data, the difficulty of these tasks may vary from trivial to very complex.

For the goals of this paper, we identify two interesting points where having the data would be of great help for performing a replicating study. So, we distinguish between the data that can be retrieved from the Internet after having identified the data source (we will call it the **"raw" data**) and the data that researchers usually use for their analysis, and that is the result of retrieving the "raw" data from the Internet, extracting the data to be analyzed, and storing it. We have labeled this more *elaborated* dataset on which researchers perform the analysis as the **processed dataset**. The output of the analysis, or at least part of it, is the research results that authors publish in the papers.

The rationale behind the decision of considering two states of the *same* information is that the transformation from "raw" data to processed dataset is usually by no means a trivial task. We can illustrate this easily with two examples that are familiar to many MSR researchers:

- Bug defect data hosted in the Bugzilla sites. The "raw" data is publicly available if access to the Bugzilla web page is provided. Researchers can just take the data from there and download to work on it. But whoever has mined Bugzilla knows the problems associated first with web scraping it (HTML templates change over time, it is time consuming, projects often block IPs as the queries are very *heavy*, etc) as well as with how to obtain structural information [4]. At the end of this tedious process, a processed dataset is available. Sometimes, an easier way to get access to the processed dataset is by requesting (and obtaining) a database dump directly from the projects. The problem is that this is not always possible and often not sufficient.
- Source code management systems. If we would like to study changes to the (programming) code of a project, the *raw* data would be the source code management system of the repository, while the processed dataset could have all large commits (that usually are due to other reasons [5]) and non-code commits filtered out. This process is non-trivial and depends largely on the selection of various parameters and choices (sliding window algorithm, definition of non-code line, etc.) that the original authors had to take.

As the amount of data in this type of studies is very large, researchers usually automatize their procedures in the form of scripts and tools. The availability of these tools/scripts is hence of key importance for the replicability of a study as the research method is, if not completely at least partially, *embedded* into them.

For all experimental papers we have tried to answer following questions:

- **Is the "raw" data publicly available?**
  As the main goal of many papers at MSR is the study of a software repository, in order to replicate a study we need access to that repository. In general, this means that researchers should have access in any form, preferably over the Internet, to the data on which the study is based. Being available in *raw* form just means that it can be somehow retrieved, even if it is *embedded* in other information and has to be consequently parsed and cleaned.
  If the "raw" data is publicly available, we further look into the paper for the time span under study. This may be a software version (for instance, version 3.0 of the Eclipse source code) or a time interval (for instance, bugs in the Eclipse Bugzilla database from September 1st 2006 to September 30th 2006). Without this information, an exact reconstruction of the dataset is not possible or results may differ significantly.
  For statistical purposes, if the source is publicly available, we have stored the name of the project. This will allow us to find the most studied projects at the MSR.
- **Is the processed dataset publicly available?**
  As noted in Figure 1, mining a software repository may include some tasks after retrieval and before the analysis. This means that usually researchers work with a *processed dataset* which is derived from the *raw* data. Obtaining this processed dataset may be a tedious and complex process, depending on the goal of the study. For this question, we investigate the papers for any reference to the dataset that authors use and that differs from the "raw" data.
- **Are tools and scripts used in the study publicly**

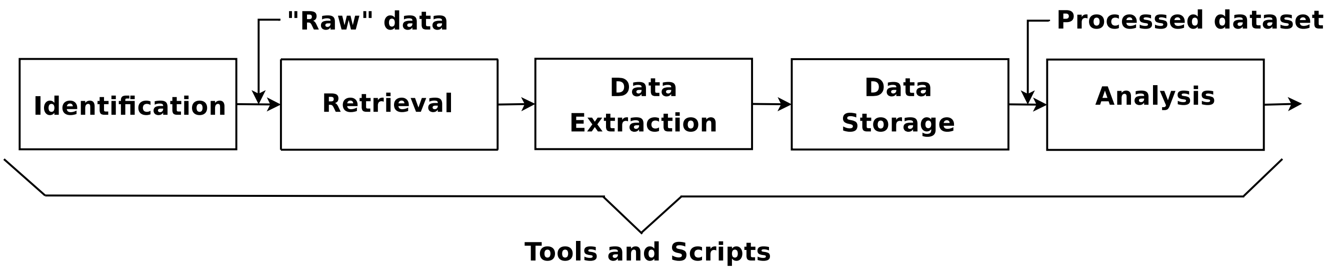


Figure 1. Typical MSR integral process: from identification of the data sources to analysis of the data.

available?

In general, researchers have automated to a great extent their studies by creating scripts or even tools that handle and analyze the data, possibly from retrieval to the final analysis. Many of the details of the method applied in all the phases of the research are implemented in those tools and scripts and even a detailed description of the process in the paper lacks the completeness or the non-ambiguity of looking at the source code.

We have studied the papers to find if a tool or a script is indicated. If so, we look if there is any reference (in the bibliography or as an URL) and have looked if it is publicly available. If no reference is given, and the name of the tool is provided, we have performed a search with a search engine. Usually we have searched first after the name of the tool itself and, in the case of too many hits, the name of the tool in combination with the author, the university or any other meaningful combination.

The goal is to find if there is a way to download the tool directly from the Internet. In this sense, tool web pages without a download link or that invite to ask the author for it are not considered as publicly available. The way in which a tool is to be found is with the source code, as a binary version hides the implementation details.

Finally, we also try to infer the version of the tool that has been used for the study, as the implementation details may vary from version to version.

The three questions that we are addressing with this study have been answered in four ways: “Yes”, “No”, “Partially” and “Not Applicable (N/A)”.

“Partially” has been chosen when many projects have been used as case studies and data is not publicly available for all of them, when the processed dataset that has been released is not complete or when the study comprehends the use of several tools and scripts and only a subset of them is available.

At last, we have labeled as “Not Applicable (N/A)” those papers that are non-experimental. Usually, these are purely theoretical or methodological papers that propose ideas and methods and that contain no case study.

III. Mining MSR for replication

A total number of 171 papers published in the proceedings of the 2004 - 2006 International Workshop on Mining Software Repositories and 2007 - 2009 Working Conference on Mining Software Repositories have been reviewed. The scope of papers under study includes long papers, position papers, short papers and “MSR Challenge” papers. The first two editions of the MSR workshop were a one-day event, while from 2006 onwards the MSR has moved to a two-day event.

Figure 2 provides a stack bar of the number of papers that have been published in the proceedings each year. In the first two years, only position papers up to five pages could be submitted, while from 2006 onwards, both long (generally up to 10 pages) or short (up to 4 pages) have been published. In the history of the MSR, we can even find a paper from 2004 that was not published in the proceedings by request of their authors.

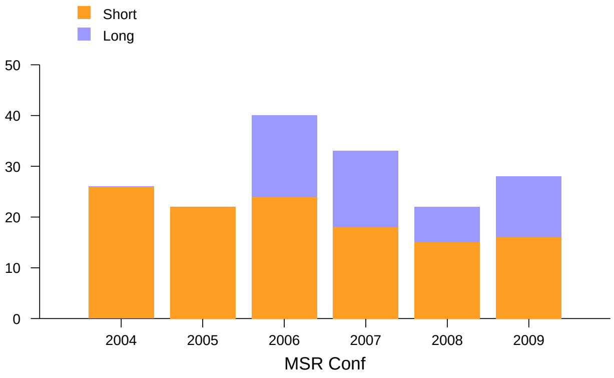


Figure 2. Number of papers by year and by length. Long papers are those with up to 10 pages, while short papers have a limit of space of up to 5 pages and include accepted submissions to the MSR Challenge.

MSR Challenge papers belong to the category of short papers, although their length has varied over time. These papers have been included in the proceedings in various forms since the creation of the challenge in 2006. So, in 2006 these papers could be 2 pages long, while in the

2007 and 2009 proceedings their maximal length was 4 pages. No MSR Challenge papers were included in the 2008 proceedings.

IV. Results

The following subsections provide an overview of the results obtained from analyzing all papers for the availability of the "raw" data, the processed dataset and the tools and scripts. First, a complete picture of the replicability of the MSR papers is given.

A. *Replicability of MSR papers*

As can be seen from Table I, papers included in the MSR proceedings are in general not very replication friendly. Out of the 154 experimental papers, only two are based on publicly available "raw" data, offer their processed dataset and provide the complete set of tools/scripts. The first of them is Germán's paper on the softChange tool from the MSR 2004 [6], while the second one is a MSR 2007 Challenge paper by Panjer on the prediction of the lifetime of Eclipse bugs [7]. Panjer uses a public processed dataset offered by the challenge organizers from the Eclipse Bugzilla and WEKA as the analysis tool.

But besides exceptions, the general rule is to find papers that are difficult, if possible, to replicate. The most frequent study published in the MSR is one that is based on publicly available "raw" data, but that does not provide the processed dataset nor the tools/scripts used for the study. 64 out of 171 papers follow this pattern. The next more frequent type of paper is the one for which "raw" data, processed dataset and tools/scripts are not available at all with 31 papers.

PRD	PDS	T&S	# Papers	% Papers
Y	Y	Y	2	1.2%
Y	Y	N	2	1.2%
Y	P	Y	1	0.6%
Y	P	P	2	1.2%
Y	P	N	2	1.2%
Y	N	Y	16	9.4%
Y	N	P	19	11.1%
Y	N	N	64	37.4%
P	N	Y	1	0.6%
P	N	N	2	1.2%
N	Y	N	2	1.2%
N	P	N	1	0.6%
N	N	Y	7	4.1%
N	N	P	2	1.2%
N	N	N	31	18.1%
N/A	N/A	N/A	17	9.9%

Table I
Statistics of availability of "raw" data, processed dataset and tools and scripts by number of papers. PRD stands for 'Public Raw Data', PDS for 'Processed dataset' and T&S for 'Tools and Scripts'

B. *Public availability of "raw" data*

Figure 3 shows the evolution of the number of papers that are based on publicly available "raw" data over the years. A first fact that can be derived from the figure is that in the last years no more non-experimental (N/A) papers have been published. In the first years of MSR, many papers on tools and methodology were published without case study. A possible explanation for this evidence is that the MSR community values in a very positive way that methods or tools presented are accompanied with a case study that show its potential and feasibility.

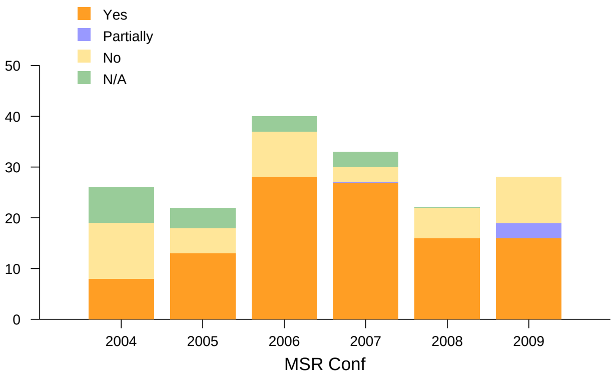


Figure 3. Evolution of papers by the public availability of the "raw" data for the case studies.

Besides that, a large use of publicly available data can be noticed, mainly from FLOSS[1] project repositories. In the year 2009, there are some studies that combine data from FLOSS projects with non-public data gathered from industrial environments, resulting only in partial availability of the data.

Among the non-public scenarios, one that has been found frequently are studies that imply monitorization of students. It should be noted that this situation *per se* does not have to imply that the data cannot be made publicly available, as the repositories could be made available after some simple privacy-preserving modifications. However, we haven't found any paper stating that this had been done.

Other non-public "raw" data correspond to those papers where the project names are not provided. We have encountered this even for the study of FLOSS projects, as it is the case in a paper where the authors show results from "one [FLOSS] community we are studying". Without any further reference, the replication of those studies is not possible. In an industrial setting, the authors sometimes point to privacy concerns to not name the projects, and fake names such as "Pocahontas" or "project X" can be found.

Finally, there are papers that use publicly available "raw" data and that link to the source, but that do not reveal the

[1] FLOSS is an acronym for Free/Libre/Open Source Software.

specific projects under scrutiny. So, randomly selecting 100 SourceForge projects or studying “[40] random GUI Java applications from SourceForge” would be of no help for researchers intending to replicate a study.

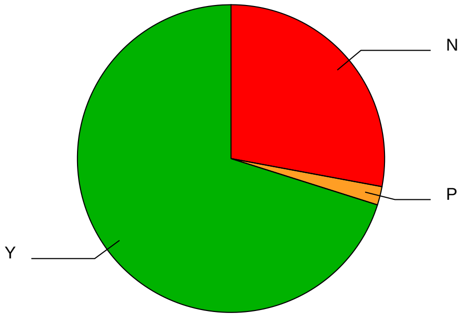


Figure 4. Distribution of papers by the public availability of the “raw” data for their case studies. Experimental papers from all MSR proceedings are considered.

Filtering out the N/A set and aggregating the data for all years, we obtain the pie shown in Figure 4. A large amount (almost three quarters) of the papers are based on publicly available data sources. For those that had publicly available “raw” data, we have written down the name of the software project that serves as data source. In this way, we have found a total number of 297 case studies performed on 174 distinct projects.

Table II contains a list of the top 20 projects that have been used as case study. Some of them may appear as the project themselves (for instance, “Apache web server” or “Evolution”) and as part of the macro-project, framework or community they belong to (for instance, “Apache” or “GNOME”). In a strict sense, they are two different sources. In the case of Sourceforge, this means that the studies performed target the platform but mine for a specific kind of project (for instance, Java projects), although some holistic analysis on the vast amount of Sourceforge projects can also be found.

It should be noted that the MSR Challenge is centered around a specific set of projects, so this fact has skewed the data shown in Table II. In this sense, PostgreSQL and ArgoUML were the focus of the MSR 2006 Challenge, Eclipse and Firefox of the MSR 2007 Challenge, Eclipse on its own of the 2008 Challenge (although no challenge paper was included in the 2008 proceedings) and GNOME was the topic of the 2009 Challenge.

Altogether, the top 10 projects have been used in 35.0% of the case studies, while the top 20 appear in 48.0% of them.

C. *Public availability of the processed dataset*

Figure 5 reveals that the amount of papers with a publicly available processed dataset is very low. In total, during the six years of the MSR, only six papers provide a means of allowing fellow researchers to work with the same data that has been used in the analysis.

Position	Source	Number
1	PostgreSQL	18
2	ArgoUML	16
3	Eclipse	15
4	Apache web server	10
5	GNOME	7
6	Linux	7
7	Sourceforge	7
8	Jedit	6
9	Mozilla	6
10	FreeBSD	5
11	GCC	5
12	Evolution	4
13	Junit	4
14	MySQL	4
15	Python	4
16	Apache	3
17	Apache Ant	3
18	Eclipse BIRT	3
19	KDE	3
20	OpenBSD	3

Table II
TOP 20 PROJECTS USED AS PUBLIC DATA SOURCES.

Of these, only two are originators of the processed dataset by themselves. The first one is Germán’s MSR 2004 softChange paper [6] that offers 6MB of a compressed database dump with Modification Requests from Evolution; while the second one is Kim et al.’s paper on clone genealogy [8] published in 2005.

In the other four cases, the authors benefited from a processed dataset that had been made available by a third party. Two of them used the Eclipse Bugzilla database dump offered for the MSR 2007 Challenge by the challenge organizers, while for the other two the data comes from a dataset published by the NASA and available in the PROMISE repository [9]. For the NASA data, although the processed dataset is publicly available, the “raw” data is not.

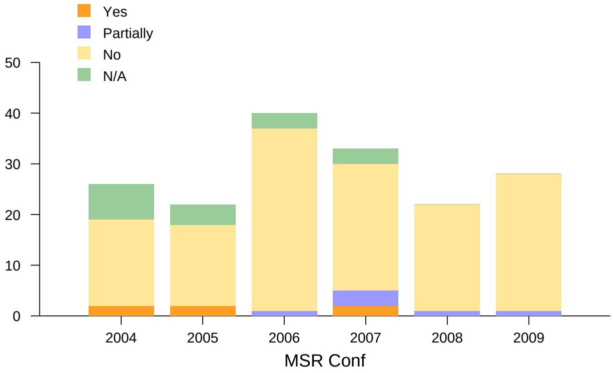


Figure 5. Evolution of papers by the publicly availability of the processed dataset.

As in the analysis of the “raw” data, no clear trend can be observed from the evolution of the availability of the processed dataset over time. If at all, there are two negative, at least regarding replicability, aspects that can be inferred from the figure:

- The amount of studies that offer or are based on publicly available processed datasets has not grown over the years. It seems that MSR researchers prefer to obtain the data in “raw” form by their own means than using other’s data.
- Some, although few, MSR authors offered in the first editions (in 2004 and 2005) their processed datasets, probably in the hope that others would re-use them or that it was the *correct* way to do. However, in more recent editions this behavior has not been observed (although partial processed datasets have been published). Interestingly enough, both authors that provided their data sources in earlier editions have published other papers in the MSR, although without publishing again their processed datasets. A possible explanation is that the MSR community does not consider the effort of doing so in a convenient manner.

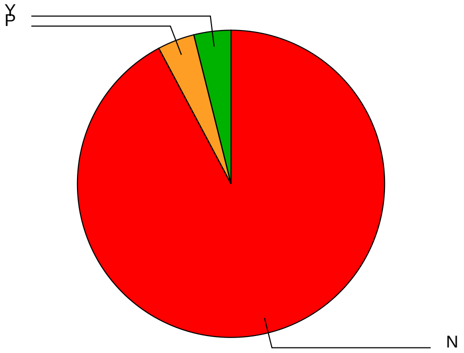


Figure 6. Distribution of papers by the publicly availability of the processed dataset. Experimental papers from all MSR proceedings are considered.

Figure 6 is the result of filtering the non-experimental (N/A) papers and displaying the distribution by the public availability of the processed dataset. Comparing it with the distribution of the “raw” data in Figure 4, the differences are significant.

One possible explication is that MSR researchers have assumed that pointing to public “raw” data is enough when it comes to other researchers trusting the analysis. This attitude is paradoxical as the MSR itself is a venue where mining is done; researchers know that the process of obtaining a processed dataset is not a simple one and that understanding how the data has been massaged is very important.

D. Public availability of tools and scripts

As a data-intensive area of research, MSR researchers have put much effort on building tools that support the retrieval and analysis of software repositories. In this sense, a large number of papers included in the MSR proceedings present tools created by the authors, many of them publicly available.

Figure 7 shows a stack diagram with the number of papers by the availability of the tools. Our expectation was that the use of publicly available tools would increase over time, as the pool of available tools grows from MSR to MSR. In spite of our assumptions, no significant trend can be identified from the figure.

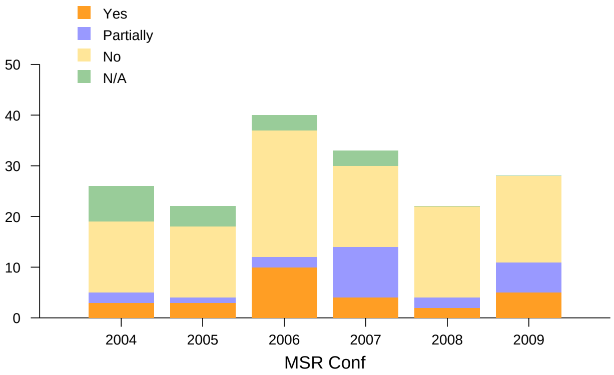


Figure 7. Evolution of papers by the public availability of tools and scripts.

According to the results shown in Figure 8, after leaving out the non-experimental (N/A) set, for around one fifth of the papers the complete set of tools and scripts to fully reproduce the study can be obtained. For around another fifth, only part of the tool/script chain is publicly available, while for a vast majority of papers no tool/script is provided.

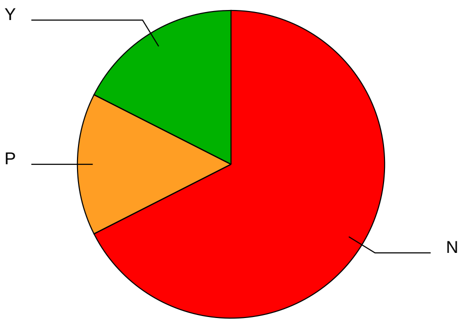


Figure 8. Distribution of papers by the availability of the tools and scripts used in the study. If tools/scripts are available (’Y’), the complete procedure used by the original authors can be obtained and used.

V. Mining Challenge

The MSR Challenge is, at least to the knowledge of the authors, a unique venue where researchers come together to present, compare and assess mining methods and tools. Since 2006, the MSR has hosted this event, providing a different submission date than for research papers and usually including accepted papers into the proceedings.

While in 2006, the challenge instructions just asked researchers to focus on a FLOSS project of their wish, since 2007 the challenge counts with two categories: a general one, that allows miners to present the usefulness of their mining tools and techniques; and a prediction one, where researchers have to predict some feature of the projects under study. In 2007 and 2008 the number of bugs during a time period for Eclipse and Mozilla had to be predicted, while in 2009 the topic was related to software growth of GNOME projects.

Together with the instructions of the challenge and the projects to be studied, the MSR Challenge organizers provide some data that may be used (or not). In the scope of this paper, some of this data may be considered as "raw" data (for instance, mirrors of the source code management systems), while other may be considered as a processed dataset (for instance, in the case of a Bugzilla database dump). All in all, as using the data provided by the organizers is not mandatory, researchers could use their own dataset. And, in general, the challenge papers do not clearly state if they have used the data sources provided by the organizers or have used the "raw" data.

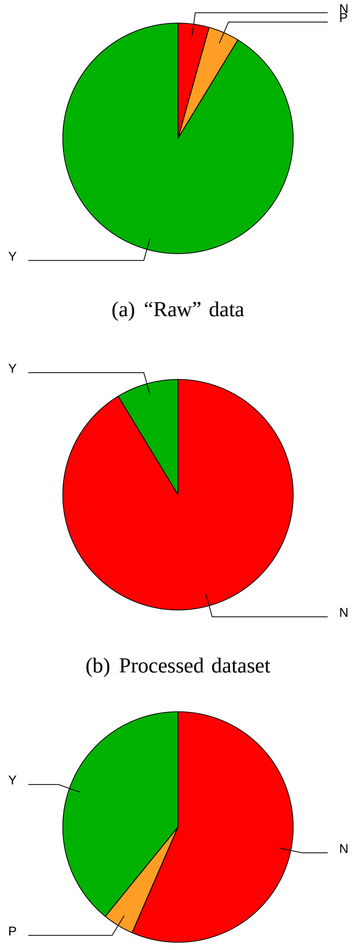
Up to date, a total number of 23 papers have been published in the MSR proceedings related to the Challenge. Distributed among the various editions in which the MSR Challenge took place, we can count 12 papers in 2006, 6 papers in 2007 and 5 papers in 2009. As already noted, no challenge papers were included in the 2008 MSR proceedings.

Figure 9 shows the public availability of "raw" data, of the processed data sets and of the tools and scripts considering only MSR Challenge papers.

While reviewing for "raw" data, we have labeled as partially available the study that considers the Bugzilla data of the 25 largest GNOME projects, not specifying the exact name of the projects under study. With that information, a replicator would have a difficult task for finding exactly the 25 of them, as it may depend on what and how bugs are counted or left out.

There is a paper which is not based on publicly available data in the MSR Challenge series. This paper studies communication on IRC channels, specifically in the #gtk IRC channel. As logs of the IRC channels are not stored and offered publicly, access to these data could only be achieved by the authors providing their processed dataset.

But, as in the case of all MSR papers, processed datasets are in general not provided. We don't know, as well, to what extent the ones offered by the organizers have been used as most challenge papers do not offer details about this. The two that we know of, as already mentioned in subsection IV-C, use the Bugzilla dump of the Eclipse project. In this case, the main cause for using the processed dataset from others is that the process that has to be followed to obtain a valuable processed dataset from the "raw" data is tedious as it involves web scraping the Bugzilla site.



(a) "Raw" data

(b) Processed dataset

(c) Tools/scripts

Figure 9. Public availability of the studied characteristics for the MSR Challenge papers. All published challenge papers are considered.

To sum up, availability of tools and scripts is more frequent in the challenge papers than for the whole MSR, but still not majority. So, it seems that the challenge is seen as a place to show how good a tool is, but not to promote its use among fellow researchers.

VI. Other interesting results

This section contains other anecdotal facts obtained while analyzing the MSR papers. This view complements the quantitative analysis performed so far providing a more complete picture of the replicability of the papers:

- No (exact) replication study at the MSR has been found in the six years under study. However, and although conceptual replication was not the scope of this study, we have found one by Tian [10], comparing a new method with an already existing one implemented in a tool.
- In around half a dozen papers we have been able to observe how authors offer very detailed information about the performance of their tools, specifying even the type of machine that was used for computation, but the tool itself is not publicly available.
- When the paper refers to *scripts* or *prototypes*, the chance of finding them publicly available is almost nil. When the paper refers to *tools*, the likelihood varies.
- There is no defined way of specifying in the paper where the data and the tool can be found. Some authors put URLs it in the body of the paper, others use footnotes, others refer to them in the bibliography providing a link and some refer to a paper or technical report in the bibliography. Finally, there are cases where the tool is named but no further reference is given.
- Many papers do not specify the time span or the software version that has been studied. Other authors research the "current version". In both cases, even having access to the "raw" data and the tool is not sufficient; the exact dates or versions should be indicated.
- Version numbers of the research tools are seldom provided in the study. As the implementation is generally *embedded* into the tools, reproducing the study becomes difficult. We have observed, in any case, that many research tools do not have version numbers and that older versions are not provided.
- While looking for tools, we have found that many of them are published in the web pages of the authors. Surprisingly, entering the URL specified in the paper has very often returned a 403/404 response (we have documented up to 17 occurrences!). Even the link for the MSR 2006 challenge data gives a 404. Researchers may not be very good web maintainers.

VII. Discussion

The following subsections offer a brief discussion about how to improve the replicability of papers for a venue such as the MSR.

A. Public available "raw" data is not enough

The MSR community has learned that especially the FLOSS community offers large amounts of publicly available data, and has widen the analyses from source code release and source code management systems to defect-tracking systems and other means of communications such as mailing lists and even IRC channels. But while the MSR community has clearly identified the benefits of the openness of the FLOSS community, it has gone the opposite direction and become more reluctant to offer similar openness.

Researchers should be aware that even if the "raw" data is completely available, they have to be very detailed about what they are studying. So, in addition to providing the time span or the version under study as already discussed, it is important that studies on a subset of the data document the subset in detail. This has to be considered, for instance, when choosing randomly projects from Sourceforge.

Another lesson that MSR researchers have to learn from the FLOSS community is that they have to tag their tools and specify what versions they are using in their research. The possibility of downloading older versions of the tools should be given. Using a source code management system with public read access would be desirable.

B. Systematic approach to replication

MSR papers should have a more systematic approach regarding the replicability of the research they present. So, as many papers have a section that specifically discusses the threats to validity, a similar one (maybe labeled "barriers to replicability") could be included[2]. A final section, just after the conclusions, has been appended to this paper discussing the replicability of this paper and pointing out that, as it happens with validity, replicability is a desirable goal but not completely achievable in many cases [11].

One of the major difficulties of this study has been to identify if papers offer additional data or tools. The way of specifying this information varies from paper to paper and in many cases, a web search has been unavoidable. The research community should standardize ways of specifying where additional information (data sources, processed dataset and tools) can be found. A possibility that is considered in this paper is adding a section at the end of the paper, just before the bibliography, as it is common with the acknowledgments.

A possible solution is to normalize how additional information could be specified, possibly offering a web page where all this information is grouped. This web page could include a link to the exact URL where the "raw" data sources may be obtained, links to the processed dataset and references to the tools, specifying the exact version of the tools that have been used.

C. Replication-specific infrastructure

But this would solve the problem partially. As we have seen from this study, the amount of 404 responses to URLs found in papers is not low. A possibility, in the same direction as the IEEE Library, and again learning from the practices that used for over a decade in the FLOSS

[2] Or it may be appended to the threats to validity section, as barriers to replicability are in fact a threat to the validity of the research.

community, would be to have a Sourceforge-like site for empirical researchers. This site could offer the services required to provide or refer to additional data and tools and ease the reuse and evolution of both. By means of the URL of the main page, access to the information of a paper could be obtained. Being so, the way to refer to this information in the paper is only a URL and could be appended, in the same way as keywords are usually provided, just after the abstract as *replication package*[12].

VIII. Conclusions

This paper has tried to shed some light on the potential replicability of all papers that have been published in the proceedings of the Mining Software Repositories. It consists of a literature review of all papers, with the aim of identifying if the data that is mined, the processed dataset used by the researchers and the tools and scripts used are publicly available. We have found that only two papers have the three characteristics and that a majority of published papers is not replication friendly.

Analyzing the characteristics by their own, we have observed that the availability of the "raw" sources is widely given, mainly due to the fact that FLOSS repositories are used. The processed datasets used by researchers are seldom provided, although we have encountered some papers in the early editions that published them. In my opinion, the MSR community does not give enough value to the availability of the processed dataset, even though it is a community that is aware of the challenges that obtaining it supposes. Finally, tools and scripts are in an intermediate position, being common that they are available over the Internet, although still a minority behavior. The findings encountered for the subset of papers that belong to the "Mining Challenge" offers similar results, although promising behaviors can be observed: reuse of processed datasets appears more often and it shows a larger share of publicly available tools.

As seen in this study, replicability tends to decrease with the age of the paper. This is because of two main reasons:

- Even if the "raw" data is publicly available, its state may change with time. This may happen, for instance, if projects that serve as case studies migrate to newer tools. Although many times there is the intention not to lose any data during the migration process, this can never be assured.
- Web pages where tools and possibly additional data is offered have a higher chance to become a lost link as time passes. In this research study we have observed many sites that return a 403 or 404 response error while trying to retrieve web pages found in the papers.

It is the opinion of the author that papers submitted to the MSR, and all those that treat with Empirical Software Engineering, should address replicability in a more formal and standardized way. In this regard, some issues are discussed in this paper and possible solutions are proposed. Among them, we encourage authors to offer a detailed description of the data being studied (including the exact time span or the versions of the software) and to indicate the specific version of the research tool used (which means that research tools should be versioned and older versions should be available as well). An agreement on a standardized way to refer to a location where additional data can be obtained should be met. A possibility could be a section at the end –as it is common for the acknowledgments– or a line at the beginning –in the same manner as it is done with the keywords. The latter is strongly tied to the necessity of having a web page with the additional information, most desirably a Sourceforge-like site that acts as a repository for this type of data and tools, and that frees researchers from maintaining infrastructure and links.

Altogether this paper has offered insight into the problems of replicating MSR papers and can serve as a good starting point to discuss future directions of how research should be performed in order to maximize replicability, and to ultimately foster the replication of research.

IX. How about the replicability of this paper?

Although this is not an experimental study, it is one that is very close to the research done in the MSR. Nonetheless, instead of mining into software repositories, the authors of this paper have actually mined the proceedings for certain patterns. That is the main reason why a major concern has been to make the study as replication friendly as possible. In this regard, and following the same method used for the rest of the MSR literature, we have tried to offer the "raw" data, the processed dataset and the scripts to fellow researchers.

- The **"raw" data** used for this study are the MSR proceedings from 2004 to 2009. The proceedings from 2004 to 2007 can be obtained from their corresponding MSR web page, while the 2008 and 2009 proceedings are only available from the ACM Digital Library.
- The **processed dataset** is composed of a structured text file (notes.txt) with an entry for each paper in the MSR proceedings. In addition to quantitative information about availability, this file contains comments and other information that support the findings, such as if the links provided in the paper are correct, where the information has been obtained (often Google searches have been done), the errors found (for instance, the 404s) and the tools used.
- The **scripts** transform the structured text used as processed dataset into database format, stores the data into a database and queries it with several other Python scripts. These scripts contain version numbers, and the version of the script used in this paper is specified in the web page.

Considering the above information, the "raw" data for this study is only partially available, while processed dataset and tools/scripts are both completely publicly available. The reason for the "raw" data being only partially available is that not all proceedings are publicly available – the ones for the years 2008 and 2009 require subscription to the ACM/IEEE Library or paying a fee per paper. We know that although access to the ACM/IEEE Library supposes no problems to university staff, it supposes a barrier to the general replicability of the study.

In accordance with the method proposed in this paper, we have added a section at the end of the paper, previous to the bibliography, that refers to the location of the "replication package".

Replication package

Links to the "raw" data, the processed dataset and the scripts that have been used to obtain the results shown in this paper can be obtained from following URL: http://gsyc.urjc.es/~grex/msr2010.

Acknowledgments

This work has been funded in part by the European Commission, through projects FLOSSMetrics, FP6-IST-5-033982, QUALOSS, FP6-IST-5-033547, and Qualipso, FP6-IST-034763. I would also like to thank the anonymous reviewers for their comments, suggestions and support.

References

[1] V. R. Basili, F. Shull, and F. Lanubile, "Building knowledge through families of experiments," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 456–473, 1999.

[2] F. J. Shull, J. C. Carver, S. Vegas, and N. Juristo, "The role of replications in Empirical Software Engineering," *Empirical Software Engineering*, vol. 13, no. 2, pp. 211–218, 2008.

[3] G. Robles, "Empirical software engineering research on libre software: Data sources, methodologies and results," Ph.D. dissertation, Escuela Superior de Ciencias Experimentales y Tecnologa, Universidad Rey Juan Carlos, 2006. [Online]. Available: http://libresoft.es/grex/phd

[4] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, "Extracting structural information from bug reports," in *Proceedings of the 2008 Working Conference on Mining software repositories*, 2008, pp. 27–30.

[5] A. Hindle, D. M. Germán, and R. C. Holt, "What do large commits tell us?: a taxonomical study of large commits," in *Proceedings of the 2008 Working Conference on Mining software repositories*, 2008, pp. 99–108.

[6] D. M. Germán, "Mining CVS repositories, the softchange experience," in *Proceedings of the International Workshop on Mining Software Repositories*, Edinburgh, Scotland, UK, 2004, pp. 17–21.

[7] L. D. Panjer, "Predicting Eclipse bug lifetimes," in *Proceedings of the 2007 Working Conference on Mining software repositories*, 2007, pp. 29–33.

[8] M. Kim and D. Notkin, "Using a clone genealogy extractor for understanding and supporting evolution of code clones," in *Proceedings of the 2005 Workshop on Mining software repositories*, 2005, pp. 17–22.

[9] G. Boetticher, T. Menzies, and T. Ostrand, "PROMISE Repository of empirical software engineering data," West Virginia University, Department of Computer Science, 2007. [Online]. Available: http://promisedata.org/

[10] K. Tian, M. Revelle, and D. Poshyvanyk, "Using latent dirichlet allocation for automatic categorization of software," in *Proceedings of the 2005 Workshop on Mining software repositories*, 2009, pp. 163–166.

[11] F. Shull, M. G. Mendonça, V. R. Basili, J. Carver, J. C. Maldonado, S. C. P. F. Fabbri, G. H. Travassos, and M. C. F. de Oliveira, "Knowledge-sharing issues in experimental software engineering," *Empirical Software Engineering*, vol. 9, no. 1-2, pp. 111–137, 2004.

[12] S. Vegas, N. Juristo, A. Moreno, M. Solari, and P. Letelier, "Analysis of the influence of communication between researchers on experiment replication," in *ISESE '06: Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*, 2006, pp. 28–37.