2014 IEEE International Conference on Software Maintenance and Evolution

ReDA: A Web-based Visualization Tool for Analyzing Modern Code Review Dataset

Patanamon Thongtanunam*, Xin Yang*, Norihiro Yoshida†, Raula Gaikovina Kula‡, Ana Erika Camargo Cruz, Kenji Fujiwara*, Hajimu Iida*

* Nara Institute of Science and Technology, Japan
{patanamon-t, kin-y, camargo, kenji-f}@is.naist.jp, iida@itc.naist.jp

† Nagoya University, Japan
yoshida@ertl.jp

‡ Osaka University, Japan
raula-k@ist.osaka-u.ac.jp

***Abstract*—ReDA(http://reda.naist.jp/) is a web-based visualization tool for analyzing Modern Code Review (MCR) datasets for large Open Source Software (OSS) projects. MCR is a commonly practiced and lightweight inspection of source code using a support tool such as Gerrit system. Recently, mining code review history of such systems has received attention as a potentially effective method of ensuring software quality. However, due to increasing size and complexity of softwares being developed, these datasets are becoming unmanageable. ReDA aims to assist researchers of mining code review data by enabling better understand of dataset context and identifying abnormalities. Through real-time data interaction, users can quickly gain insight into the data and hone in on interesting areas to investigate. A video highlighting the main features can be found at: http://youtu.be/_fEoTRRas0U**

***Keywords*—Modern Code Review, Mining software repository, Visualization tool**

I. Introduction

Software code review refers to inspection of the source code by developers, other than the author, who have appropriate knowledge of the software. Code review is an established method for reducing defects and improving software quality. Nowadays, Modern Code Review (MCR) [1], an informal, lightweight and tool-based code review methodology has been widely used and put into development regularly in both industrial and OSS projects. Many tools such as Gerrit[1], ReviewBoard[2], Rietveld[3] are available to support MCR. ReDA focus on Gerrit which supports code reviews for projects using Git code repository system. Gerrit facilitates a review of code changes and management of patchsets (set of changes) which developers expect to merge into the code repository.

Due to the increasing use of MCR in several large OSS projects, massive volumes of code review data have been recorded. Their histories have been captured and these datasets are publicly available online. Extracting knowledge from these datasets has produced promising research with the goal of improving the software quality and software development process. Recently, many studies have used code review datasets to understand and improve both review effort [2]–[5] and review quality [6], [7]. However, a raw code review dataset is generally imperfect since the data collection process in each support tool varies in methodology, accuracy, and degree of automation [8]. Reliability of the data depends on the variety of inputs and operations, for instance, input from default values of the systems, mistaken input of users, and data missing or removed data (cleaned up). As Kalliamvakou et. al. [9] reported that while publicly available data from support tools is a rich source on mining, various potential perils should also be taken into consideration. Therefore, to avoid such perils, researchers must understand the context and quality of a dataset to decide what further data pre-processing and cleansing is required.

To facilitate researchers understanding of a code review dataset, we developed a prototype of a data visualization tool named ReDA – Review Data Analyzer. ReDA is a lightweight and interactive web-based visualization tool that enables users to quickly understand large and complex code review data. ReDA leverages both review activity and human factors for the analysis by extracting proprieties available from Gerrit and visually presenting them to users. Throughout this paper, we illustrate the use of ReDA using review history from Android Open Source Project (AOSP) captured from October 2008 to January 2012 with 11,632 reviews [10].

Related Work. It is generally accepted that visualization can help developers to understand and discover knowledge from a software repository. Minelli et al. [11] mined software repositories of apps and presents the mined data to understand evolution of apps. Yongpisanpop et al. [12] also presents that interacting with big data in a bugs reporting system through visualization encourage developers to provide feedback as a request for quality improvement. While bug reporting is closely related to MCR, prior to ReDA there was no visualization tool available explicitly for MCR.

In this paper, we demonstrate how ReDA assists researchers in three ways: 1) Extracting and defining the dataset, 2) Showing basic statistical analyses, and 3) Indicating interesting and potentially major problems in the project.

II. ReDA Overview

ReDA is a web application to enable distributed platform independent access and a familiar, easy to use interface. It is implemented with HTML5 and JavaScript for the front-end and python Django for the server backend. Figure 1 shows an architectural overview of ReDA. MCR repository datasets (e.g. from Gerrit system) are exported in JSON and imported to ReDA database. From the database, ReDA can easily access

[1]https://code.google.com/p/gerrit/
[2]https://www.reviewboard.org/
[3]https://code.google.com/p/rietveld/

1063-6773/14 $31.00 © 2014 IEEE
DOI 10.1109/ICSME.2014.106

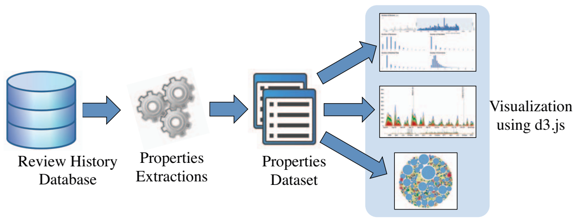


Fig. 1: An architectural overview of ReDA

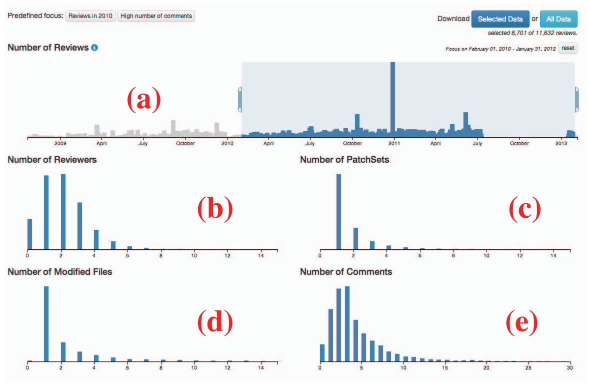


Fig. 2: An overview of Review Statistic page

and extract properties of the project and then visualizes them using d3.js[4] which is a javascript library for manipulating data and graphically presents results. ReDA also provides summaries of the extracted properties.

To support researchers comprehensive understanding the datasets, ReDA provides and visualizes the properties of projects from three MCR perspectives: 1) **Review** perspective shown in Review Statistic page, 2) **Process** perspective shown in Activity Statistic page, and 3) **Human** perspective shown in Contributor Activities page.

A. Review Statistic

The goal of the Review Statistic page is to present an overview of reviews that AOSP developers have performed. An overview of Review Statistic page is shown in Fig. 2. On this page, users can find a large graph on the top Fig. 2(a) showing how many reviews have been created per week. The four small graphs below Fig. 2(b)-(e) show the distribution of properties which are the number of reviewers, the number of patchsets, number of modified files and number of comments, respectively. These graphs are interactive. Users can focus the data on any graph by dragging a slider over any desired section. Then, every graph will be changed corresponding to the selection area. For example, if users want to see distributions of the metrics of reviews created in 2011, they can drag over the Number of Reviews graph as the blue area shown in Fig.2(a). Then, the graphs of Fig.2(b)-(e) will show the distributions corresponding to the selected portion. Multiple graphs can be selected so users can compare and understand relationships among properties in different areas of the data.

[4]http://d3js.org/

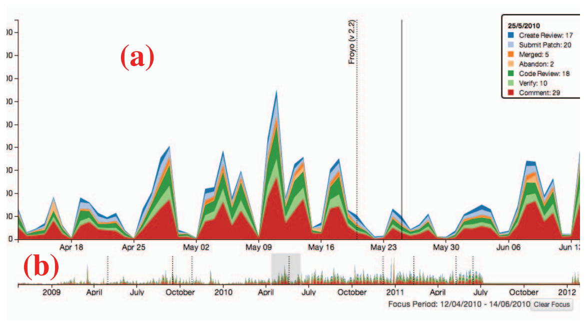


Fig. 3: An overview of Activity Statistic page

B. Activity Statistic

The goal of Activity Statistic page is to present the overall of activity in the process of MCR. Figure 3 shows an overview of this page where the large graph on the top Fig. 3(a) shows the number of activities performed in Gerrit on a daily basis. The small graph below Fig. 3(b) is for focusing and zooming on activities history in a specific period. ReDA extracts the activities in this process into seven different types: 1) Create review, 2) submit patch, 3) code review, 4) verify, 5) merge changes, 6) reject changes, and 7) give a comment. For zooming detail, users can drag a selection over the graph of Fig. 3(b). Then, the graph of Fig. 3(a) will display the selected portion. The exact number of activities in each day will be shown in the description box at the top right when the cursor is hovering over the graph 3(a). ReDA also shows the date for Android release version[5] via a dashed line to show how these activities are before/after the release date. For example, as shown in Fig. 3(a), the number of activities before releasing the Froyo version is higher than the activities after the release.

C. Contributor Activities

The goal of Contributor Activities page is to present the viewpoint of human in MCR. Figure 4 shows an overview of this page where Fig. 4(a) shows contributors graph, Fig. 4(b) describes chart legend and Fig. 4(c) is the visualizing menu for this graph. In Fig. 4(a), ReDA plots a bubble chart where each bubble represents individual contributor. For each contributor, ReDA extracts the number of four main activities: 1) Create Review, 2) code review, 3) verify and 4) comment. The size of bubble represents the total number of their contributions. The larger bubbles present the more contributions. The color of the bubble represents the combination of activity types that they have done. The color reference is described in the panel in Fig. 4(b). For instance, a blue bubble means they did all four types of contributions, and an orange bubble means they only did a code review. Users can see the exact number of contributions by hovering the cursor over the desired bubble. Then, a description box will appear next to the cursor. Moreover, users can filter the contributors by selecting the minimum number of contributions at the drop-down box in Fig. 4(c). When the number is selected, the graph will be automatically re-loaded.

[5]based on http://en.wikipedia.org/wiki/Android_(operating_system)

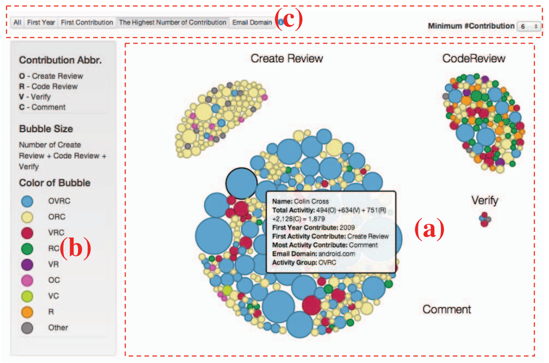


Fig. 4: An overview of Contributors Activities page

ReDA can separate contributors into groups to view different aspects. For example shown in Fig. 4(a), the contributors are grouped by the activity they have done the most. Users can select the type of group at the menu at Fig. 4(c). This can visualize the relationship between activities they have done (represented by color) and the type of grouping. For example, from Fig. 4(a), we found that most of contributors, who create a review as the activity they have done the most, had no verify activity (yellow and pink bubbles).

III. Illustrative Usage Scenario

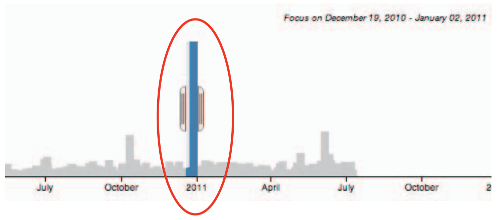
In this section, we demonstrate the use of ReDA to identify abnormalities in the dataset and characteristics of activities in AOSP. Moreover, we examined the dataset according to the findings to uncover their causes.

At the Review Statistic page, the number of reviews being created in the end of 2011 is markedly higher than other periods over the time as shown in Fig. 5(a). Similarly, the Activity Statistic page shows that in the same period, the number of created reviews is unusually high as shown in Fig. 5(b). In particular, the number of reviews created is high on 23/12/2010 and the number of reviews abandoned is high on 29/12/2010. Following this evidence, we investigate the cause of this phenomenon using the following two questions:

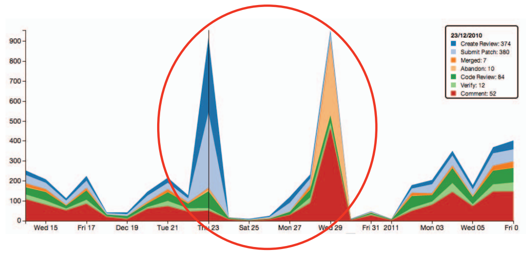
- Q1: What reviews were created and abandoned in this period?
- Q2: Is there a review in the dataset similar to these reviews?

To answer Q1, we first filtered and downloaded the dataset in the Reviews Statistic page by dragging over the highest bar on the Number of Review graph. We found that from 374 reviews[6] created on 23/12/2011, there are 356 reviews (98%) that were abandoned on 29/12/2010. Moreover, these reviews were created by the same developer at consecutive times. These reviews are the changes for supporting Nvidia Tegra family. The properties' values of these reviews can be summarized as `#Reviewers` $\leq$ 1, `#Patchset` $=$ 1, `#Modified files` $>$ 0, and `#Comment` $=$ 1. We scanned through their comments to find the reason of creating and abandoning these reviews. Unfortunately, there is no explanation from the owner.

[6]Result for Q1: http://reda.naist.jp/static/findings/ReDA_finding_1.htm



(a) Unusual number of created reviews in Review Statistic Page



(b) Unusual number of activities in Activity Statistic Page

Fig. 5: An abnormality identified using ReDA

Only automatically generated messages recorded that these changes were abandoned. We can answer Q1 that *the reviews were created and abandoned in this period are those without performing code review*.

To better understand this phenomenon and answer Q2, we selected reviews that have the same metric values for the whole captured period. We simply selected these reviews by dragging a selection over the bars on the small four graphs in the Review Statistic page. There are 1,305 reviews[7] returned and 924 (89%) of them are abandoned. We then grouped reviews that occurred with the same pattern as found in Q1 i.e. reviews created by the same developer at consecutive time. We then manually scanned through the comments to find an explanation. For the result, we can identify 633 reviews that have the same pattern including reviews found in Q1. Moreover, we found that this phenomenon happens more than once a month. From their comments we found that 18% of these reviews are abandoned because these changes were already fixed, had a new version, or the code is obsolete; and 11% of them are abandoned due to the mistakes of developers such as a lack of Git and Gerrit knowledge, writing a bad commit message, and accidentally uploading the changes. However, the rest of these reviews have no explanation. We conjecture that the cause of these reviews can be one of the causes we have found previously. For instance, it is possible that developers accidentally uploaded the changes if we consider that the creation of reviews is continuous (the difference of creation time is in seconds). A further investigation is needed to explain this clearly. At this stage, we can answer Q2 that *there are reviews that are similar to reviews we found in Q1 and they occasionally occurred over the project*.

Besides the findings we have discussed, we also discovered other abnormalities and characteristics. In Table I, we summarizes the findings that we have identified using ReDA. This

[7]Result for Q2: http://reda.naist.jp/static/findings/ReDA_finding_2.htm

TABLE I: Summary of findings are discovered using ReDA.

	Summary	Evidences in ReDA	Findings
F1:	10% of all reviews were created not for code review.	In Review Statistic and Activity Statistic page, there are reviews created and abandoned with unusual numbers at the end of year 2010.	There are 924 reviews that were created and abandoned without performing code review. There are 55% of these reviews were abandoned without discussion. Furthermore, some of them were created by mistakes such as uploading changes that already fixed, accidentally uploading changes.
		In Review statistic page, there are reviews that do not contain changes (No modified file).	There are 217 reviews that have no modified file. From their commit messages, we found that 93% of them are branch merging requests which are not related code reviews.
F2:	Review history in AOSP is missing.	In Number of Reviews graph of Review Statistic page, there is no reviews created from July 2011 to January 2012.	We found that on this period, Git and Gerrit servers were down for a while. This was reported in Google forum [13] by Jean-Baptiste Queru, the software engineer for Google AOSP.
F3:	AOSP developers usually do code review on weekdays.	In Activity Statistic page, the number of activities are periodically low.	We found that the days with low number of activities are usually Saturday and Sunday.
F4:	Most of main contributors of AOSP are the members of Google and Android teams.	In Contributor Activities, most of large bubbles have blue color.	We found that most of large blue bubbles have email domain google.com and android.com. These contributors also have a number of activities higher than the others in the same group of the first year of contribution.

table describes evidences shown in ReDA and findings from these evidences.

IV. Experience with ReDA.

From our experience, we mined MCR dataset of AOSP to better understand human aspects. We identified main contributors by profiling from code review history [2] and analyzed important roles in MCR using social network analysis (SNA) [3].

From our studies, we found that main contributors are AOSP and Google people. This result is corroborated by finding F4 of ReDA described in Table I. Additionally, we also used the Contributors Activities page validating the result of SNA, concretely when we separated a group of verifier and non-verifier. According to this, we believe that ReDA provides evidence that can be investigated to improve the code review process in the future.

V. Summary & Future Work

In this paper, we present ReDA a visualization tool for understanding code review datasets generated through MCR support tools. We demonstrate the use of ReDA on a large, complex OSS project and find code review abnormalities. Using ReDA, we found that there is noise and incorrect information hidden in the dataset. This shows that researchers must be aware and avoid this peril and carefully refine the dataset before using it.

Future work. We will develop ReDA to provide real-time code review data "dashboard" system. This would enable users to monitor current status of code reviews and quickly indicate problems in the project and code review process. Other important aspects of code reviews will also be developed to expand code review viewpoints. Furthermore, ReDA is not limited to only researchers. It is also useful for industrial project managers to monitor projects and quickly identify problems; or perform lessons learned for process improvement activity or for use in satisfying CMMI process quality requirements. We will develop a portable version of ReDA as well as make it compatible for other MCR support tools.

Acknowledgments

We are thankful to Assoc. Prof. Daniel Port for his valuable suggestions and constructive comments.

References

[1] A. Bacchelli and C. Bird, "Expectations, Outcomes, and Challenges of Modern Code Review," in *Proc. of ICSE'13*, 2013, pp. 712–721.

[2] R. G. Kula, C. C. A. Erika, N. Yoshida, K. Hamasaki, K. Fujiwara, X. Yang, and H. Iida, "Using Profiling Metrics to Categorise Peer Review Types in the Android Project," in *Proc. of ISSRE'12*, 2012, pp. 146–151.

[3] X. Yang, R. G. Kula, C. C. A. Erika, N. Yoshida, K. Hamasaki, K. Fujiwara, and H. Iida, "Understanding OSS Peer Review Roles in Peer Review Social Network (PeRSoN)," in *Proc. APSEC'12*, 2012, pp. 709–712.

[4] V. Balachandran, "Reducing Human Effort and Improving Quality in Peer Code Reviews using Automatic Static Analysis and Reviewer Recommendation," in *Proc. ICSE'13*, 2013, pp. 931–940.

[5] P. Thongtanunam, R. G. Kula, A. E. C. Cruz, N. Yoshida, and H. Iida, "Improving Code Review Effectiveness through Reviewer Recommendations," in *Proc. of CHASE'14*, 2014, pp. 119–122.

[6] S. Mcintosh, Y. Kamei, B. Adams, and A. E. Hassan, "The Impact of Code Review Coverage and Code Review Participation on Software Quality Categories and Subject Descriptors," in *Proc. of MSR'14*, 2014, pp. 192–201.

[7] M. Beller, A. Bacchelli, A. Zaidman, and E. Juergens, "Modern Code Reviews in Open-Source Projects : Which Problems Do They Fix ? Categories and Subject Descriptors," in *Proc. of MSR'14*, 2014, pp. 202–211.

[8] A. Mockus, "Engineering Big Data Solutions," in *Proc. of FOSE'14*, 2014, pp. 85–99.

[9] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, "The Promises and Perils of Mining GitHub Categories and Subject Descriptors," in *Proc. of MSR'14*, 2014, pp. 92–101.

[10] K. Hamasaki, R. G. Kula, N. Yoshida, C. C. A. Erika, K. Fujiwara, and H. Iida, "Who does what during a Code Review ? An extraction of an OSS Peer Review Repository," in *Proc. MSR'13*, 2013, pp. 49–52.

[11] R. Minelli and M. Lanza, "SAMOA – A Visual Software Analytics Platform for Mobile Applications," in *Proc. of ICSM'13*, 2013, pp. 476–479.

[12] P. Yongpisanpop, H. Hata, and K. Matsumoto, "Bugarium: 3d Interaction for Supporting Large-Scale Bug Repositories Analysis," in *Proc. of ICSE Companion'14*, 2014, pp. 500–503.

[13] J.-B. Queru, "Google Groups Disccusion." [Online]. Available: https://groups.google.com/forum/#!searchin/android-contrib/gerrit$20down/android-contrib/m5FPkdI3ImQ/vL9_61Dof6AJ