Gerrit Software Code Review Data from Android

Murtuza Mukadam Concordia University Montreal, QC, Canada m_mukada@concordia.ca Christian Bird Microsoft Research Redmond, WA, USA cbird@microsoft.com Peter C. Rigby Concordia University Montreal, QC, Canada peter.rigby@concordia.ca

Abstract—Over the past decade, a number of tools and systems have been developed to manage various aspects of the software development lifecycle. Until now, tool supported code review, an important aspect of software development, has been largely ignored. With the advent of open source code review tools such as Gerrit along with projects that use them, code review data is now available for collection, analysis, and triangulation with other software development data. In this paper, we extract Android peer review data from Gerrit. We describe the Android peer review process, the reverse engineering of the Gerrit JSON API, our data mining and cleaning methodology, database schema, and provide an example of how the data can be used to answer an empirical software engineering question. The database is available for use by the research community.

I. INTRODUCTION

The tools used to support software projects have provided a rich source of data for software engineering research. For example, common tools include source code management systems, build systems, bug databases, and test infrastructures. The results of such research have included insight into software development practices (e.g. [5]) and tools to aid practitioners (e.g. [6]). However, there are few mining scripts and datasets available for studying tool supported peer review.

Software inspection has been an engineering "best practice" for over 35 years [3]. Email based Open Source Software (OSS) peer review has been extensively studied and been shown to be effective [7]. It is important to understand how tool supported peer review is impacting the effectiveness of this engineering practice. In this data paper, we describe how we mine the Gerrit [4] peer review system to extract reviews done by Android developers. We provide a dataset that includes information about which software changes are reviewed (and implicitly, which are not), who typically looks at such changes, how long reviews take, and what types of discussions and feedback are given during code review.

The paper is structured as follows: we describe the Gerrit based peer review process used by the Android project (section II), the source of the data, the methods we used to collect the data along with challenges and limitations (section III), a description of the data schema (section IV), and finally we show an example of how such data can be used to answer questions relating to review practices and discuss future avenues of research with the data (section V).

II. DESCRIPTION OF PROJECT AND DATA

Android is an operating system developed with a goal to create real-world products which improves the experience for users using mobile and tablet devices. It was initiated by Android Inc., and was bought by Google in 2005. Open Source Software and was initiated by a group of companies known as Open Handset Alliance in 2007, which is led by Google [1]. The Android community uses the free web based software code review tool Gerrit [4]. We downloaded a total of 19k reviews from Gerrit.

Gerrit is integrated with git and serves as a barrier between developers' private repositories and the official, centralized Android source tree [2]. Developers make local changes and then submit these changes for review. Reviewers make comments via the Gerrit web interface. For a change to be merged into the Android source tree, it must be approved and verified by a senior developer. Android is an example of a review-then-commit policy [7] that has additional change approval steps [2]:

- 1) "Verified" Before a review beings, someone must verify that the change merges with the current master branch and does not break the build. In many cases, this step is done automatically.
- "Approved" While anyone can comment on the change, someone with appropriate privileges and expertise must approve the change.
- "Submitted/Merged" Once the change has been approved it is merged into Google's master branch so that other developers can get the latest version of the system.

The example in Figure 1 illustrates a review in Android.¹ A Gerrit review begins when the owner (Shuo Gao) posts a patch to be reviewed. Reviewers are assigned (Jeffrey Brown, Christophe Bransiec, etc.) so that they can take part in the reviewing of the patch uploaded by the owner. Unassigned reviewers can also make comments. Reviewers can provide comments on individual lines that have changed (2 comments) or they can provide general comments (Jean Baptiste Queru comments "Patch Set 1: Verified"). Reviewers can approve (Christophe Bransiec gives a value +1) or reject (Jeffrey Brown, a value of -2) the uploaded patch. The bot (Deckard Autoverifier) comments, "Patch Set 2 is verified". A patch set encapsulates details regarding the author, committer and also the inline comments made by the reviewers. Multiple patch sets can be uploaded during a review (2 patch sets have been uploaded).

¹https://android-review.googlesource.com/#/c/41591/

Change-Id:	Ibf60045cf8a673dec30	87eb7a2743063cc20	e7eb 💼 Com	mit Message				rennann	
Owner	Shuo Gao		(KLO	[KLOCWORK] [SQL] : fixes null dereference					
Project	Q platform/frameworks/base		This	This patch fixes a null dereference issue reported by					
Branch	master		kloc	klocwork. Change-Id: lbf60045cf8a673dec3087eb7a2743063cc20e7eb Author: Ovidiu Beldie ≺ovidiux.beldie@intel.com≻					
Topic			Auth						
Uploaded	Aug 16, 2012 11:	Aug 16, 2012 11:20 PM			Signed-off-by: Ovidiu Beldie <ovidiux.beldie@intel.com> Signed-off-by: Christophe Bransiec <christophex bransiec@intel.com<="" td=""></christophex></ovidiux.beldie@intel.com>				
Updated	Feb 6, 2013 10:43 PM		Sign	Signed-off-by: Shuo Gao <shuo.gao@intel.com></shuo.gao@intel.com>					
Submit Type	Always Merge	Always Merge		Signed-off-by: Jack Ren <jack.ren@intel.com></jack.ren@intel.com>					
Status	Review in Progres	SS	Auth	or-tracking-B2:	34345				
Reviewer		Code-Review	Verified						
Deckard Auto	verifier		~						
Jeffrey Brown	1	×							
Christophe B	ransiec	+1							
Bruce Beare		+1							
Johan Redes	tig								
Jack Ren									
Jean-Baptiste	e Queru								
Depender	union.								
Dependen	icies								
	listory: Base	-							
Id Version F		1170004500117050	2daEbE20E1 (altilog						
Patch Set	1 238326187C222								
Patch Set	1 23e32eta/C222								
ld Version F ▶ Patch Set ♥ Patch Set	2 bbbcd73c38ea	5e89a4b77a0d2e14	d74243f4eod3 (gitik	, :s)					
Patch Set	23e32eta7c222 bbbcd73c38eat Christophe Bran	5e89a4b77a0d2e14 siec <christophe< td=""><td>d74243f4ecd3 (gitile ex.bransiec@intel</td><td>, s) .com> Aug 3, 201;</td><td>2 8:48 A</td><td>м</td><td></td><td></td></christophe<>	d74243f4ecd3 (gitile ex.bransiec@intel	, s) .com> Aug 3, 201;	2 8:48 A	м			
Patch Set Patch Set Patch Set Autho Committe	23e32eta7c222 bbbcd73c38eat Christophe Bran Shuo Gao <shuo< td=""><td>5e89a4b77a0d2e14 siec <christophe o.gao@intel.com</christophe </td><td>d74243f4ecd3 (gitile ex.bransiec@intel i> Sep 5, 2012 2:</td><td>, .com> Aug 3, 201; 54 AM</td><td>2 8:48 A</td><td>М</td><td></td><td></td></shuo<>	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com</christophe 	d74243f4ecd3 (gitile ex.bransiec@intel i> Sep 5, 2012 2:	, .com> Aug 3, 201; 54 AM	2 8:48 A	М			
Patch Set Patch Set Patch Set Autho Committe Parent(s)	2363268376222 bbbcd73c38ea Christophe Bran Christophe Bran Shuo Gao <shuo b307ceb43fa96ebo<="" td=""><td>5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is4851c20278a484</christophe </td><td>d74243f4ecd3 (gitik ex.bransiec@intel a> Sep 5, 2012 2:8 4469c9269 Merge</td><td>, .com> Aug 3, 201; 54 AM "SearchWidget: p:</td><td>2 8:48 A assing A</td><td>M PP_DATA to voice</td><td>e intent"</td><td></td></shuo>	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is4851c20278a484</christophe 	d74243f4ecd3 (gitik ex.bransiec@intel a> Sep 5, 2012 2:8 4469c9269 Merge	, .com> Aug 3, 201; 54 AM "SearchWidget: p:	2 8:48 A assing A	M PP_DATA to voice	e intent"		
Patch Set Patch Set Patch Set Autho Committe Parent(s) Download	2 bbbcd73c38eat Christophe Bran r Shuo Gao <shuc b307ceb43fa96etc , checkout chec</shuc 	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is9851c20278a434 kout pull che</christophe 	ad74243f4ecd3 (gilik ex.bransiec@intel a> Sep 5, 2012 2:: 4469es269 Merge erry-pick patch	, .com> Aug 3, 201; 54 AM "SearchWidget: p:	2 8:48 A assing A	M PP_DATA to voice	e intent"		
 Patch Set Patch Set Autho Committe Parent(s) Download 	2363268/2222 bbbcd73c3868 Christophe Bran Shuo Gao <shuc b807ceb49fa9666="" chec="" chec<="" checkout="" td=""><td>5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is9851a20278a434 kout pull che platform/fram</christophe </td><td>d7424314ecd3 (gitik ex.bransiec@intel a> Sep 5, 2012 2: 4469c9269 Merge erry-pick patch neworks/base 41</td><td>, .com> Aug 3, 201; 54 AM "SearchWidget: p; .591/2 👔</td><td>2 8:48 A assing A</td><td>M PP_DATA to voice</td><td>e intent"</td><td></td></shuc>	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is9851a20278a434 kout pull che platform/fram</christophe 	d7424314ecd3 (gitik ex.bransiec@intel a> Sep 5, 2012 2: 4469c9269 Merge erry-pick patch neworks/base 41	, .com> Aug 3, 201; 54 AM "SearchWidget: p; .591/2 👔	2 8:48 A assing A	M PP_DATA to voice	e intent"		
 Patch Set Patch Set Patch Set Autho Committe Parent(s) Download File 	234328876222 bbbcd73c38884 Christophe Bran Shuo Gao <shuc b307ceb43fa9686="" chec="" checkout="" download="" path<="" repo="" td=""><td>5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is4851c20278_4934 kout pull che platform/fram</christophe </td><td>d7424314ecd3 (gitik ex.bransiec@intel i> Sep 5, 2012 2: 44469c9269 Merge erry-pick patch neworks/base 41</td><td>2 25) 26) 26 AM *SearchWidget: pi 25) 20) 20) 20) 20) 20) 20) 20) 20</td><td>2 8:48 A assing A Size</td><td>M PP_DATA to voice</td><td>e intent"</td><td></td></shuc>	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com is4851c20278_4934 kout pull che platform/fram</christophe 	d7424314ecd3 (gitik ex.bransiec@intel i> Sep 5, 2012 2: 44469c9269 Merge erry-pick patch neworks/base 41	2 25) 26) 26 AM *SearchWidget: pi 25) 20) 20) 20) 20) 20) 20) 20) 20	2 8:48 A assing A Size	M PP_DATA to voice	e intent"		
 Version F Patch Set Patch Set Autho Committe Parent(s) Download File Con 	29832887/2222 bbbcd73c38888 Christophe Bran Shuo Gao <shuc b307ceb43fa9686="" chec="" checkout="" download="" message<="" mmit="" path="" repo="" td=""><td>5e89a4b77a0d2e14 siec <christophe o.gao@intel.com istessic20270_454 kout pull che platform/fram</christophe </td><td>d7424314ecd3 (gilik ex.bransiec@intel a> Sep 5, 2012 2:: 4446969269 Merge emy-pick patch seworks/base 41</td><td>2 25) 26) 26) 26) 26) 26) 26) 26) 26</td><td>2 8:48 A assing A <i>Size</i></td><td>M PP_DATA to voice Diff Side-by-Side</td><td>e intent"</td><td></td></shuc>	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com istessic20270_454 kout pull che platform/fram</christophe 	d7424314ecd3 (gilik ex.bransiec@intel a> Sep 5, 2012 2:: 4446969269 Merge emy-pick patch seworks/base 41	2 25) 26) 26) 26) 26) 26) 26) 26) 26	2 8:48 A assing A <i>Size</i>	M PP_DATA to voice Diff Side-by-Side	e intent"		
 Patch Set Patch Set Patch Set Autho Committe Parent(s) Download File Com M core 	2000 2000 2000 2000 2000 2000 2000	5689a4b77a0d2e14 siec <christophe o gao@intel.com istesic202702494 kout pull che platform/frea</christophe 	d7424314ecd3 (gitik ex.bransiec@intel l> Sep 5, 2012 2:: 4449052259 Merge erry-pick patch meworks/base 41	 .com> Aug 3, 2012 54 AM .591/2 2 <i>Comments</i> via 2 comments 	2 8:48 A assing A <i>Size</i> +1, -1	M PP_DATA to voice Diff Side-by-Side Side-by-Side	e intent" Unified		
 Patch Set Patch Set Patch Set Autho Committe Parent(s) Download File Cone M core 	20620817C22 bbb07338ee r Christophe Bran r Shuo Gao <shuu android="" ava="" b07eebs12546es="" chec="" checkout="" datal<="" download="" message="" mmit="" path="" repo="" td=""><td>5689a4b77a0d2e14 siec <christophe o.gao@intel.com istes1c20276a494 kout pull che platform/fras</christophe </td><td>d7424314ecd3 (gillik xx.bransiec@intel i>> Sep 5, 2012 2:: 444959245 Merge prry-pick patch neworks/base 41 teQueryBuilder.js</td><td>2 2 2 2 2 2 2 2 2 2 2 2 2 2</td><td>2 8:48 A assing A <i>Size</i> +1, -1 +1, -1</td><td>M PP_DATA to voice Diff Side-by-Side All Side-by-Side</td><td>e intent" Unified Unified All Unified</td><td></td></shuu>	5689a4b77a0d2e14 siec <christophe o.gao@intel.com istes1c20276a494 kout pull che platform/fras</christophe 	d7424314ecd3 (gillik xx.bransiec@intel i>> Sep 5, 2012 2:: 444959245 Merge prry-pick patch neworks/base 41 teQueryBuilder.js	2 2 2 2 2 2 2 2 2 2 2 2 2 2	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1	M PP_DATA to voice Diff Side-by-Side All Side-by-Side	e intent" Unified Unified All Unified		
 Version F Patch Set Patch Set Autho Committe Parent(s) Download File Con M core 	Zeszera (ZZZ Zeszera (ZZZ Zeszera (ZZZ) Christophe Bran Christophe Bran Shuo Gao <shuo <shuo="" android="" datal<="" download="" gao="" java="" message="" mmit="" path="" shuo="" td="" zepo=""><td>5689a4b77a0d2e14 siec <christophe 0.gao@intel.com istestac20276a494 kout pull che platform/free base/sqlite/SQLi</christophe </td><td>d7424314ecd3 (gilik ix.bransiec@intel i>Sep 5, 2012 2: 444595265 Merge prry-pick patch beworks/base 41 teQueryBuilder.ja</td><td>com>Aug 3, 201: 54 AM 'SearchWidget: p: 591/2 Comments Va 2 comments</td><td>2 8:48 A assing A <i>Size</i> +1, -1 +1, -1</td><td>M PP_DATA to voice Diff Side-by-Side Side-by-Side All Side-by-Side</td><td>e intent" Unified Unified All Unified</td><td></td></shuo>	5689a4b77a0d2e14 siec <christophe 0.gao@intel.com istestac20276a494 kout pull che platform/free base/sqlite/SQLi</christophe 	d7424314ecd3 (gilik ix.bransiec@intel i>Sep 5, 2012 2: 444595265 Merge prry-pick patch beworks/base 41 teQueryBuilder.ja	com>Aug 3, 201: 54 AM 'SearchWidget: p: 591/2 Comments Va 2 comments	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1	M PP_DATA to voice Diff Side-by-Side Side-by-Side All Side-by-Side	e intent" Unified Unified All Unified		
Version F Patch Set Autho Committe Parent(s Downloac File Con M core Somments	29620997222 206007338eet Christophe Bran Shuo Gao <shuo 200="" 300="" sh<="" shuo="" td=""><td>568944b7780d2e14 siec <christophe gac@intel.com is4851c202754134 kout pull che platform/free base/sqlite/SQLi</christophe </td><td>d7424314ecd3 (gilik ix.bransiec@intel)> Sep 5, 2012 2: 444595269 Merge prry-pick patch beworks/base 41 teQueryBuilder.js</td><td>com>Aug 3, 201; 54 AM 'SearchWidget: p: 591/2 Comments Va 2 comments</td><td>2 8:48 A assing A <i>Size</i> +1, -1 +1, -1</td><td>M PP_DATA to voice Diff Side-by-Side Side-by-Side All Side-by-Side Expand Recent</td><td>e intent" Unified Unified All Unified Expand All</td><td>Collapse All</td></shuo>	568944b7780d2e14 siec <christophe gac@intel.com is4851c202754134 kout pull che platform/free base/sqlite/SQLi</christophe 	d7424314ecd3 (gilik ix.bransiec@intel)> Sep 5, 2012 2: 444595269 Merge prry-pick patch beworks/base 41 teQueryBuilder.js	com>Aug 3, 201; 54 AM 'SearchWidget: p: 591/2 Comments Va 2 comments	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1	M PP_DATA to voice Diff Side-by-Side Side-by-Side All Side-by-Side Expand Recent	e intent" Unified Unified All Unified Expand All	Collapse All	
Version F Patch Set Patch Set Patch Set Autho Committe Parent(s Downloac File Con M core Comments ean-Baptiste	2 bbbd733684 2 bbbd733684 2 bbbd733684 7 Christophe Bran r Shuc Gao <shuu)="" 2="" android="" b307eeb3fa3681="" chec="" checkout="" datal<="" download="" message="" mnit="" o="" path="" repo="" td="" vjava=""><td>568944b778042e14 siec <christophe gag@intel.com ss4951c20276454 kout pull che platform/frea base/sqlite/SQLi</christophe </td><td>d742434ed3 (piliked) xx.bransiec@intel >> Sep 5, 2012 2: 4446453265 Morge <u>mry-pick patch </u> tecQueryBuilder.je code style is fine,</td><td>the change merge</td><td>2 8:48 A assing A <i>Size</i> +1, -1 +1, -1</td><td>M PP_DATA to voice Diff Side-by-Side Side-by-Side All Side-by-Side Expand Recent</td><td>e Intent" Unified Unified All Unified All Unified Expand All Aug</td><td>Cottapse All 25 11:31 AM</td></shuu>	568944b778042e14 siec <christophe gag@intel.com ss4951c20276454 kout pull che platform/frea base/sqlite/SQLi</christophe 	d742434ed3 (piliked) xx.bransiec@intel >> Sep 5, 2012 2: 4446453265 Morge <u>mry-pick patch </u> tecQueryBuilder.je code style is fine,	the change merge	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1	M PP_DATA to voice Diff Side-by-Side Side-by-Side All Side-by-Side Expand Recent	e Intent" Unified Unified All Unified All Unified Expand All Aug	Cottapse All 25 11:31 AM	
 Version F Patch Set Patch Set Authon Committe Parent(s) Downloac File Con M core Comments ean-Baptiste huo Gao Up 	2 Sebectario222 2 Sebectario222 2 Sebectario222 7 Christophe Bran Shuo Gao <shuo 2="" 2<="" android="" batorswistariseta="" chec="" checkout="" datai="" ioaded="" patch="" queru="" set="" td="" vijava=""><td>5e89a4b77a0d2e14 siec <christophe o.gao@intel.com istestic2027e1494 kout pull che platform/fran base/sqlite/SQLi</christophe </td><td>d7424314ed3 (piliki xx.bransiec@intel I>> Sep 5, 2012 2: 144655265 Morge pry-pick patch neworks/base 41 IteQueryBuilder.ja</td><td>(com> Aug 3, 2012 54 AM "SearchWidget: pr 591/2 Comments wa 2 comments the change merge</td><td>2 8:48 A assing A <i>Size</i> +1, -1 +1, -1</td><td>M PP_DATA to voice Diff Side-by-Side All Side-by-Side Expand Recent</td><td>e Intent" Unified Unified All Unified All Unified All Unified Second All</td><td>Coltapse All 25 11:31 AM ap 5 2:59 AM</td></shuo>	5e89a4b77a0d2e14 siec <christophe o.gao@intel.com istestic2027e1494 kout pull che platform/fran base/sqlite/SQLi</christophe 	d7424314ed3 (piliki xx.bransiec@intel I>> Sep 5, 2012 2: 144655265 Morge pry-pick patch neworks/base 41 IteQueryBuilder.ja	(com> Aug 3, 2012 54 AM "SearchWidget: pr 591/2 Comments wa 2 comments the change merge	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1	M PP_DATA to voice Diff Side-by-Side All Side-by-Side Expand Recent	e Intent" Unified Unified All Unified All Unified All Unified Second All	Coltapse All 25 11:31 AM ap 5 2:59 AM	
Veld Version F Patch Set Patch Set Patch Set Autho Committe Parent(s Download File Con M core Comments ean-Baptiste huo Gao Up ohan Redes	2 Sectora CCC 2 Sectora CCC 2 Sectora CCC 7 Christophe Bran 7 Shuo Gao <shuo 7="" <shuo="" gao="" shuo="" shuo<="" td=""><td>568940740d2e14 siec <christophe > gao@intel.com istsssca2072a434 source [puil feature] base/sqliter/SQLi 1: Verified The c - inline comment]</christophe </td><td>d742434ed3 (ptill xx.bransiec@intel >> Sep 5, 2012 2: 4445ss2c8 Merge my-pick pach neworks/base 41 teQueryBuilder.jp</td><td>the change merge</td><td>2 8:48 A assing A <i>Size</i> +1, -1 +1, -1</td><td>M PP_DATA to voice Diff Side-by-Side Side-by-Side Expand Recent </td><td>e intent" Unified Unified All Unified All Unified Expand All Aug Se</td><td>Collapse All 25 11:31 AM ap 5 2:59 AM ap 5 6:18 AM</td></shuo>	568940740d2e14 siec <christophe > gao@intel.com istsssca2072a434 source [puil feature] base/sqliter/SQLi 1: Verified The c - inline comment]</christophe 	d742434ed3 (ptill xx.bransiec@intel >> Sep 5, 2012 2: 4445ss2c8 Merge my-pick pach neworks/base 41 teQueryBuilder.jp	the change merge	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1	M PP_DATA to voice Diff Side-by-Side Side-by-Side Expand Recent	e intent" Unified Unified All Unified All Unified Expand All Aug Se	Collapse All 25 11:31 AM ap 5 2:59 AM ap 5 6:18 AM	
Vald Version F Patch Set Patch Set Patch Set Autho Committe Parentis Download File Con M core Comments ean-Baptiste ihuo Gao Up ohan Reddes teckard Auto	Constant of the second se	568540740d2e14 siec <christophe > gao@intel.com 5455520274a44 kout pull che platform/fra base/sqlite/SQLI 1: Verified The c - I: Verified The c</christophe 	a742434ed3 (gilik ix.bransiec@intel >> Sep 6, 2012 2: 	so) com> Aug 3, 2011 54 AM "SearchWidget: pr 551/2 Comments Va 2 comments the change mergen	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1 es and	M PP_DATA to voice Off Side-by-Side Side-by-Side Expand Recent	e intent" Unified Unified All Unified Expand All [Aug Se Se Oc	Collapse All 26 11:31 AM ap 5 2:59 AM ap 5 6:18 AM 11 11:48 PM	
Veld Version F Veld Version F Version F Veld Version F Version F Veld Veld Veld Veld Veld Veld Veld Veld	Comparison of the second	568940740d2e14 siec <christophe gao@intel.com ist851202078434 base/sqlite/SQLi 1: Verified The c inline comment) 2: Verified The c</christophe 	dr22434edf (glill) xx branslec@intel >> Sep 5, 2012 2: 4449:s226 Merge Herevorks/base 41 teQueryBuilder, js code style is fine, s an automated r he app.	essage from the	Size +11,-1 +1,-1 	M PP_DATA to voice PP_DATA to voice PP_DATA to voice PD/ff Side-by-Side Side-by-Side Expand Recent	e Intent" Unified Unified All Unified All Unified All Unified Expand All Se Se Occ	Cottapse All 25 11.31 AM ap 5 259 AM ep 5 618 AM 11 1.48 PM 1.148 PM	
Veld Version F Veld Version F Version F Veld Version F V	Constant of the second of	568940740d2e14 siec <christophe gao@intel.com sssssize0275444 base/sqlite/SQLi 1: Verified The c Inline comment) 2: Verified This is -Review 2: Fix the 2: Code-Review</christophe 	dr2434ecd (gills) ix bransiec@intell ix bran	com> Aug 3, 201; 54 AM "SearchWidget p S91/2 Comments Comments the change merge	2 8:48 A assing A size +11, -1 +1, -1	M Daff Side-by-Side Side-by-Side Expand Recent	e Intent" Unified Unified All Unified All Unified All Unified Se Se Occ Dec	Cottapse All 251 131 AM ap 5 2 59 AM ap 5 6-18 AM 11 11 48 PM : 10 9:00 PM 11 5 28 AM	
Veld Version F ► Patch Set Veltch Set Autho Committe Parent(s Download File ► Con M core Comments ean-Baptiste ihuo Gao Up ohan Redes teckard Auto effreg Brow ihristophe Bi incce Beare	Constant of the second of	se8949740d2e14 siec <christophe gao@intel.com tistsicazora.eix kout pull che platform/fran tistsicazora.eix hille commenti) 2. Verified The ci- inline commenti) 2. Verified This i -Review 2 Fix t - Code-Review 1 Jeff.</christophe 	dr22434edf (gille x branslec@intel >> Sep 5, 2012 2: Hetsszer Morgo mry-pick [path] everorks/base 41 code style is fine, is an automated in the app. 1 (1 comment) i guess that this	ess) com> Aug 3, 201; 54 AM "SearchWidget pr 591/2 Comments wa 2 comments the change merger nessage from the is really a questio	2 8:48 A assing A +1, -1 +1, -1 es and . Deckard	M PP_DATA to voice Diff Diff Side-by-Side Side-by-Side Expand Recent .	e intent" Unified Unified All Unified All Unified Aug Ss Ss Oc Dec Dec Unified	Collapse All 25 1131 AM 35 5 259 AM 11 1148 PM 10 9 00 PM 11 5 28 AM 11 5 28 AM	
Veld Version F Patch Set Autho Committe Parch Set Autho Committe Parch Set Downloac File Comments ean-Baptiste ihuo Gao Up ohan Redes leckard Autho effrey Brown ihrisophe Bi muce Beare effrey Brown	Constant of the second se	seeseko7raodze14 siec <christophe gao@intel.com issesscapraeseko platform/frame base/sqlite/SQLI 1: Verfiled The c - inline commenti) 2: Verfiled The c - inline commenti) 2: Verfiled The c - inline commenti 2: Code-Review Review+1 Jeff</christophe 	ar22434exd (gma xx bransec@intellex >> Sep 5, 2012 2: >> Sep 5, 20	set of the change merger is a comments and the change merger is search with the change merger is search the change merger is really a question.	2 8:48 A assing <i>A</i> +1, -1 +1, -1 +1, -1 n	M Diff Diff Side-by-Side Side-by-Side Expand Record	e intent" Unified Unified All Unified All Unified Aug Se Se Oc Dec Dec Dec Dec	Coltapse All 25 11 31 AM sp 5 259 AM pp 5 618 AM t11 48 PM t1 0 9:00 PM t1 5 28 AM 17 72 PM 5 10 49 AP	
Veld Version F ≥ Patch Set 2 Patch Set Autho Committe Parent(s Download Filia > Con M core Comments ean-Baptiste ihuo Gao Up ohan Redes Veckard Autho effrey Brown thristophe Bri ince Beare effrey Brown ************************************	Constant of the second se	5689640740d2e14 silec <christophe gao@intel.com tistesicastratesi kouti puli che piatform/fran base/sqitte/SQLI 1: Verified The c</christophe 	ar22434ecd (gmail kx.bransiec@intellect vs.Sep 5, 2012 2: vs.Sep 5, 2012 2: vmspick_patch neworks/base 41 teQueryBulder,p code style is fine, s an automated r he app. +1 (1 comment) 1 guess that this	com> Aug 3, 201; 54 AM "SearchWidget p 551/2" Comments Comments the change merge nessage from the is really a questio	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1 es and .	M Dirf Side-by-Side Side-by-Side Expand Recent	e intent" Unified All Unified All Unified All Aug Se Se Se Oc Dec Dec Dec Dec Fet	Collapse All 25 11:31 AM pp 5 256 AM pp 5 618 AM 11 148 PM 21 5 28 AM 17 7 27 PM 6 10 43 PM	
Velat Version F Patch Set Autho Committe Partch Set Autho Committe Partch Set Download File Comments constant Comments constant Comments constant	Comparison of the second	5689640740d2e14 siec <christophe gao@intel.com tistesica272+ack platform/fran base/sqliterSQLI 1: Verified The c inline comment) 2: Verified The c inline comment) 2: Verified The s Review+1 Jeff</christophe 	4722434ecd (generation) xx bransiec@intellev >> Sep 5, 2012 2: ##59328 Mego my-pick patch accords / base 41 teQueryBulder, patch teQueryBulder, pat	com> Aug 3, 201; 54 AM "SearchWidget p "SearchWidget p Comments Comments Comments is really a questio is really a questio ws IllegalArgument	2 8:48 A assing A size +1, -1 +1, -1 es and .	M Diff Side-by-Side Side-by-Side Expand Recent	e intent" Unified All Unified All Unified All Grand Se Se Oc Dec Dec Dec Dec Fet	Collapse All 25 11:31 AM pp 5 2:59 AM sp 6 6:18 AM 11 1 48 PM 11 1 528 AM 11 7 7 27 PM 56 10:43 PM	
Velat Version F Patch Set Patch Set Autho Committe Parent/set Paren	Constant of the second of	sedesto7faod2e14 siec <christophe gao@intel.com stessicao27a-ask kouti puit i che platform/fear base/sqlite/SQLI 1: Verified The c linline comment) 2: Verified The s Review-2 factorm/fear Review-1 deff hecks whether d Heck, ng Kirking</christophe 	ar22434ecd (g) and (g) ar22434ecd (g) are considered (g) and (g) are considered (g) and (g) are considered (x solutions and a set of the set	2 8:48 A assing A Size +1, -1 +1, -1 ts and . Deckard n tExcepti	M Ont Side-by-Side Side-by-Side Expand Recent	e intent" Unified Unified All Unified All Unified Aug Se Occ Dec Dec Dec Dec Dec Dec Dec Dec Dec	Collapse All 25 11 31 AM up 5 259 AM up 5 259 AM up 5 259 AM 11 1.48 PM 21 15 28 AM 17 7 27 PM 26 10 43 PM	
Version F Valch Set Patch Set Autho Committe Parent(s Download Download Download Comments aan-Baptiste huo Gao Up ohan Redes eekard Auto affrey Brown hristophe Brown hath Sec	Comparison of the second	568949760d2e14 siec <christophe gao@untel.com sistosizzeraak kout puti che platform/fran base/sqlite/SQLI 1: Verified The to inline comment) 2: Verified The to inline comment) 2: Verified The to Review-1 Jerr. Review+1 Jerr.</christophe 	4722434ecd (g88 x branslec@intel >> Sep 5, 2012 2: tesses 5, 2012 2: tesses 6, 2012	ess) com>Aug 3, 201; 54 AM SearchWidget p SearchWidget p Comments Comments Var 2 comments the change merget in essage from the Is really a question we sillegalArgumen ull when mTables	2 8:48 A assing A <i>Size</i> +1, -1 +1, -1 es and . Deckard n tExcepti is null to	M Diff Diff Side-by-Side Side-by-Side Expand Recent	e intent" Unified Unified All Unified Aug Ss Ss Ss Ss Ss Ss Ss Ss Ss Ss Ss Ss Ss	Collapse All 251131 AM pp 5.259 AM sp 5.618 AM 111 148 PM 10 9:00 F 15 628 AM 117 727 PM 6 10 43 PM	

Fig. 1. Gerrit Android Review Number 41591

III. EXTRACTION METHOD

Android review data is stored in Gerrit.² We were able to avoid screen-scraping by observing how Gerrit web pages are constructed. Gerrit works by initially sending a web page skeleton and some Javascript to the browser. The Javascript then makes a number of web requests back to the Gerrit server and requests information about code reviews, which is returned in JSON format. The page DOM is then modified by the Javascript to display the code review information. When we developed our script, the Gerrit REST API provided limited information. However, the current Gerrit API provides an interface to JSON formatted review data.³ This JSON data must still be parsed.

We used the developer tools within the Chrome web browser to inspect these web requests (inspecting header fields and POST data) to the Gerrit server along with the responses in an effort to reverse engineer the types of web methods available and the structure of the JSON data returned by the requests. We also determined which fields in the displayed web pages corresponded to what fields within the JSON. The JSON returned was fairly complex, deep, and redundant (it was not uncommon for a single JSON response for a code review to exceed 50 kilobytes). In addition, many web requests were needed to obtain all information about an individual review.

v currentPatchSetId: {changeId:{id:41591}, patchSetId:2} ▼ changeId: {id:41591} id: 41591 patchSetId: 2 dependsOn: [] messages: [{key:{changeId:{id:41591}, uuid:AAAI6X///9M=}, author: neededBy: [] v patchSets: [{id:{changeId:{id:41591}, patchSetId:1}, revision:{id Ø0: {id:{changeId:{id:41591}, patchSetId:1}, revision:{id:23e32e createdOn: "2012-08-17 03:20:48.036000000" draft: false vid: {changeId:{id:41591}, patchSetId:1} ▼ changeId: {id:41591} id: 41591 patchSetId: 1 > revision: {id:23e32efa7c2221f2e86458cf1f7e563da5b53051} ▶ uploader: {id:1014559} Fig. 2. JSON Response from the server

One review might contain many rounds of *patch sets* (an author may submit one set of changes, get feedback, submit a set of revised changes, etc.). Obtaining the information for each patch set requires an additional web request, and gathering the reviewer comments for each file within each patch set requires yet another. Thus, a review might require over twenty to thirty individual web requests. In an effort to avoid overloading the Gerrit server (and also avoid our IP being blacklisted from the site), we throttled our mining by delaying one second between requests.

We developed a Python script that made use of the various web methods and extracted the relevant data from the JSON responses. We also created a database schema based on the information returned from the server and the data was stored in a Microsoft SQL Server database for later analysis. To enable broad use of the data, we provide an SQL Server database backup file as well as a simple XML dump of the data.⁴

Data Extraction Details and Example

We reverse engineered the JSON requests to get the "ChangeDetailService" and "PatchDetailService". In Figure 2, we show a snippet of the JSON returned when we sent a request for the ChangeDetailService for the review in Figure 1. We store the raw data from each JSON request, so that we can re-process the data without sending requests to the server.⁵ Our Python script then extracts data from the JSON into a database. For example, the patch id (["re-sult"]["patchSet"]["id"]["patchSetId"]=2), change type (['re-sult']['patches']='A'), lines added (['result']['patches']=19), and lines removed (['result']['patches']=0).

A. Challenges and Limitations

We describe some of the data limitations and some challenges we overcame while cleaning anomalies from the data. We hope that as this dataset becomes more widely used for answering empirical software engineering research questions, other challenges and limitations will be identified and removed from the data.

Challenge: Gerrit JSON API

The Gerrit JSON API is the only way to get all information from Gerrit. While the API is intended for public use, it is

²https://android-review.googlesource.com

³https://gerrit-review.googlesource.com/Documentation/rest-api.html

⁴Script and data is available at https://github.com/mmukadam/gerrit-miner.git
⁵Please contact us for a dump of the raw data



Fig. 3. Number of Reviews started per Month

not formally defined and there is no documentation for it. As we discussed above we reverse engineered the JSON requests and responses. However, the API can change between versions of Gerrit and the location and names of services can also change. For example, while mining Android, the location of the "ChangeDetailService" was moved from

/gerrit/rpc/ChangeDetailService to
/gerrit_ui/rpc/ChangeDetailService

Challenge: A Bot and other Anomalies

Looking for anomalies in the data, we noticed that a bot called "Deckard Autoverifier" was involved in 1.8k reviews. Qualitative analysis revealed that the bot was responsible for ensuring that the change merged with the master branch without conflict and that it did not break the build - it was "Verified". Since the bot is responsible for automatically verifying new changes, we expected there to be one verification for each patch set. However, there are 19k reviews and the bot is involved in only 1.8k. A mailing list discussion⁶ revealed that the "Deckard Autoverifier" cannot verify inter-project dependencies, so many verifications must be done manually. For example, on AOSP "Jean-Baptiste Queru" manually verifies all new changes. Since Queru does many manual verifications, he will have commented on an artificially large number of review. Depending on the goal of future analysis, verifications with no other comments may be tagged or removed.

Challenge: Collecting all Reviews

Reviews are identified by an id number; however, not every review number contains a valid id. We download all reviews between review number 1 to 51750, which resulted in 19k reviews. Figure 3 plots the number of reviews per month. While the number of reviews can fluctuate drastically in a given month [7], there is evidence on the Android mailing lists that the Gerrit database has been cleaned at various points in time, removing stalled reviews, but leading to missing data. For example, many reviews are missing from August 2011 until the start of January 2012. A solution would be to regularly mine the Android Gerrit data.

⁶https://groups.google.com/forum/#!msg/android-contrib/cEFSGewsqUQ/ umd5FKrv4_QJ

IV. DATA STORAGE AND SCHEMA

The database schema is depicted in Figure 4. In general, each table has an *Id* column that is unique for each entry in the table (e.g., the *Review* table has *ReviewId*, the *Person* table has *PersonId*). Foreign key relations are indicated by the presence of a column in one table that has the same name as the primary key of another table.

Reviews

The *Review* table contains information about the review itself. This includes the review identification number used by Gerrit (the primary key), the person that created the review and typically made the change (*OwnerId*), the creation time (*CreatedOn*), the last time that any activity occurred on the review (*LastUpdatedOn*), a one line description of the change (*Subject*) along with a description (*message*), the project that the review is for, and the branch within git that the change was made on. The *Status* can be "open", meaning that the review is active and the change has not yet been accepted,"merged", meaning that the review has passed and the change has been merged into the codebase, and "abandoned", indicating that the review has not passed and is no longer active.

People

Many tables include references to people (reviewers, authors, *etc.*) through the use of a *PersonId*. The *Person* table maps this id to the person's *Name* and *Email* address. We have observed that some automated system accounts also add information to reviews. For example, one "bot" adds comments to a review that describe changes to the review. These accounts can be identified due to the *IsBotAccount* being set to 1.

Patch Sets

A change for review is made up of a set of files that correspond to a commit. In Gerrit parlance, this is called a patch set. As an author responds to feedback, he may submit multiple patch sets until the final patch set is accepted. The *PatchSet* table includes information including whether this is the first, second, third, *etc.* patch set for a review (*PatchSetNumber*), the number of files in the patch set (*NumberOfFiles*), when it was created (*CreatedOn*), and the revision within the git repository that contains the versions of the files in the patch set (*GitRevision*).

Patch Set Files

Each file within a patch set has an entry in the *PatchSetFile* table. This includes the path of the file (*Path*), how many lines were added and deleted, and the *ChangeType*, which indicates if the file was added, removed, or modified. We do not store this in the database. It is easy to obtain this with the *GitRevision* from the corresponding *PatchSet* entry.

Comments

The information about each comment is in the *Comment* table. This includes the text of the comment (*Message*), when the comment was made (*WrittenOn*), who wrote the comment (*AuthorId*), and which patch set the comment is relevant to



Fig. 4. Database Schema



Fig. 5. Number of reviews submitted per day of the week over the last half of 2012 in Android.

(*PatchSetId*). Comments can refer to a particular location within a file in the patch set, in which case the file path (*Path*), line within the file (*LineNumber*), and whether the comment refers to the version of the file prior to or after the change (*Side*) are indicated. Otherwise, these fields will be NULL.

Approvals

Each review is given certain points(-2,-1,0,1,+2) based on whether reviewers judge that it should be accepted or rejected. This information is stored in the *Approval* table, with an entry for each person involved in a review. This is the only table that doesn't have a single field primary key, as the *ReviewId* and the *PersonId* uniquely identify the entry and act as a conjugate primary key. The information in this table indicates whether the reviewer has signed off on the review (*ReviewedStatus*) and/or verified that the change does not cause problems (*VerifiedStatus*) and stores when these occurred (*ReviewedWhen* and *VerifiedWhen*).

V. FUTURE WORK

In a forthcoming paper in preparation for submission, we have used this data to quantitatively characterize code review

in Android and compare Android code review to review in other contexts. However, here we present a simple illustration of the types of questions that can be answered using this data. We want to know how developers apportion their work over the course of their work week.

Figure 5 shows boxplots that describe the distribution of the number of code reviews submitted per weekday over the past six months in the Android project. Using t-tests (since the submission numbers per day follow normal distributions) we determined that there is no statistical difference between Monday, Tuesday, Wednesday, Thursday, and Friday. However, they all show a statistically significant increase over both Saturday and Sunday and Saturday has more than Sunday to a statistically significant degree. Thus, one can reasonably conclude that the contributors to Android work during the week and take weekends off.

We are currently using this data set, other OSS datasets, and datasets from software firms to understand how different software development environments affect peer review practices. Other research avenues include using this new dataset to improve models of defect prediction, identifying attributes of changes that lead to many comments from reviewers or many iterations of change submission, and characterizing review patterns of developers who join software projects.

REFERENCES

- Android. Android Open Souce Project. http://http://source.android.com/ index.html.
- [2] Android. Submitting patches. http://source.android.com/source/ submit-patches.html.
- [3] M. Fagan. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*, 15(3):182–211, 1976.
- [4] Gerrit. Web based code review and project management for git based projects. http://code.google.com/p/gerrit/.
- [5] A. Hindle, D. M. German, and R. Holt. What do large commits tell us?: a taxonomical study of large commits. In *MSR*, 2008.
- [6] R. Holmes and A. Begel. Deep intellisense: a tool for rehydrating evaporated information. In MSR, 2008.
- [7] P. C. Rigby, D. M. German, and M.-A. Storey. Open Source Software Peer Review Practices: A Case Study of the Apache Server. In *30th ICSE*, pages 541–550, 2008.