Background Study on Developers' Information Needs

Research shows that interesting insights can be obtained from combining these sources [**?**], [**?**]. However, researchers face various problems in analysing these sources due to challenges in mining the sources and managing the extracted data. We surveyed the literature that focus on studying developers information needs from different external sources, as shown in **??**. We present the challenges and benefits researchers find using various sources.

A. Methdology

For this analysis, we selected the relevant papers considered in a recent systematic literature review (SLR) discussing the developers information needs in program comprehension tasks [**?**]. This SLR groups the studies in people-centric and technology-centric related work. Our selection process was focused on 29 papers from the technology-centric group, concerning the analysis of different sources such as bug reports, newsgroups, and mailing lists. We further complemented this initial set of studies, by including works that focus on studying developers information needs from other platforms such as mobile app stores (e.f., user reviews data) platforms and Quora. This step was performed on *Google Scholar*, by following the same criteria used in the the aforementioned SLR study [**?**]. In particular, we included the study if it focuses in part or whole on software developer information needs related to software development and includes empirical evidence. We excluded if the study is a review, survey, tool study, older than 15 years, not peer-reviewed, and not in english, As a result, we added 23 additional papers, resulting in a total of 52 papers. As our focus is on diversity of sources rather than a deep overview of a particular source, we excluded the studies analyzing same project from same source. The full list of identified papers is reported in **??**.

B. Observations

Based on the corpus of relevant literature (**??**), we identified the sources that capture developer information needs, as listed in **??**: *Q&A Site*, *Mailing list*, *Newsgroup*, *Bug Report* or *User Review*. Please note that each analyzed study can rely on multiple sources or various projects in a source. For instance, *e.g.*, Zagalsky *et al.* used developer information needs data from Mailing lists and Stack Overflow and thus mentioned in categories *Q&A* and *Mailing lists*[**?**] whereas Sharif and Buckley report a study involving six different mailing lists (Apache BSF, Eclipse JDT, Element Construction Set, Eboard, SwingWT, Reciprocate)[**?**]. We argue that the identified source categories represent a diverse list for researchers to extract and analyze data to capture developers information needs. The focus of the analysis was to uncover the characteristics, challenges and problems addressed in each category of sources in order to shed some light on the current research methodologies.

TABLE I
DATA SOURCES RESEARCHERS USE TO UNDERSTAND DEVELOPER NEEDS

Category	Platform / Project	# papers	Total per category
Q&A Site	stackoverflow.com	25	
Q&A Site	quora.com	2	**27**
Mailing list	Apache BSF Site	3	
Mailing list	Eclipse JDT	3	
Mailing list	Element Construction Set	1	
Mailing list	Eboard	1	
Mailing list	SwingWT	1	
Mailing list	Reciprocate	1	
Mailing list	R	1	
Mailing list	Qt	1	
Mailing list	Ubuntu	1	**5**
Bug Report	Eclipse	1	
Bug Report	Mozilla	1	**1** (the same paper)
Newsgroup	Java Swing Framework	2	**2**
User Review	Google Play Store	6	
User Review	Apple App Store	1	
User Review	Windows Phone Store	1	
User Review	Blackberry App Store	1	**8**

Q&A Sites. Even when documentation on a project or software component is available, it is still hard to apply the information directly to the software development tasks [**?**]. Therefor, developers turn to Q&A sites as they are a valuable resource to ask peers [**?**] and find accurate answers [**?**]. Stack Overflow is one of the most popular Q&A sites for developers [**?**] with millions of questions, answers, and comments on a wide range of topics [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. The data from Stack Overflow is made officially accessible through various channels (Data Dump, Data Explorer, API), which eases the process of gathering massive amounts of data from Stack

Overflow. The data from Stack Overflow contains HTML tags, source code and other metadata that need to be pre-processed (e.g., with data cleaning) prior to any automated analysis. Despite the fairly easy extraction of data from Stack Overflow, researchers face recurrent challenges in selecting relevant data and preprocessing (noise removal) it for their studies [**?**], [**?**]. To address the data selection challenge, multiple approaches based on tag selection [**?**], [**?**], [**?**], keyword selection [**?**] or using the whole dataset [**?**] have been proposed by previous studies. To address the data preprocessing challenge, various tools and services have been proposed to clean the data but not all studies report the precise preprocessing steps and tools used.

Quora is another Q&A platform used by developers, but not many software engineering research studies investigated this source to capture developers information needs. One of the reasons can be the difficulties in extracting data from Quora compared to Stack Overflow. Indeed, Quora site does not provide any public API or Data Dump to obtain relevant data. Additionally, its web site is difficult to scrape because of limitations for not logged-in users and extensive use of Javascript in the frontend [**?**], [**?**].

Observation 2.1 Q&A sites, led by Stack Overflow, are an often used and valuable online resource to investigate developer information needs about a particular technology or topic. The convenient access to data from Stack Overflow has enabled numerous research studies to explore it compared to the usage of Quora for similar purposes.

Mailing lists. Mailing lists are often used in large, long-lived open source software projects [**?**] such as Linux[1] and Apache[2]. With the rise of social Q&A sites, user support activities tend to shift away from mailing lists [**?**]. Nevertheless, mailing lists are a significant source of information on open source projects [**?**], [**?**] and are still actively used in many. Sharif *et al.* characterize mailing lists as "*the backbone of open source communications*" [**?**], pointing to the fact that many long-lived open source software projects used and still use mailing lists as the primary communication tool between developers. Mailing lists have the advantage that the communication records often span multiple years and are related to one specific project. This allows researchers to uncover the evolution aspects of developer information needs and how they differ in multiple stages of a project. The research using this source is primarily focused on analyzing developer communication [**?**], [**?**], [**?**], [**?**] and categorizing the information [**?**], [**?**], [**?**].

Four out of the six observed studies extract mailing list data manually, which is a time-intensive task and yields little data. Unfortunately, the content in the mails is often bloated with footers, automatically generated content or stack traces that require cleaning from the dataset [**?**], [**?**]. Additionally, mailing list data consists only of unstructured text (*i.e.*, no tags or assigned topics) which makes it difficult to split it semantically or to extract specific topics.

[1]https://lkml.org/

[2]https://mail-archives.apache.org/mod_mbox/

Observation 2.2 *Mailing lists* are a valuable resource to research developer communication, information seeking behavior, and evolution aspects of a project. The unstructured nature of emails and lack of supporting tools in mining tasks poses challenges for researchers.

Bug Reports. Bug tracking systems are essential for enabling technical communication between developers or external contributors from the community [**?**]. Bug reports offer two different ways to help developer information needs. First, a developer fixing a bug can find detailed information about the environment, configuration and circumstances that lead to the bug, providing crucial information to fix the bug. Second, a user posting a bug report can obtain valuable feedback from developers on how to integrate or use the product or software component. Breu *et al.* show that fixing a bug is a collaborative process as the active participation of the developer and user is essential for successfully resolving the bug [**?**]. Baysal *et al.* identified the need of reducing information overload present on bug tracking system [**?**] whereas Panichella *et al.* utlized the communication from bug tracking systems to satisfy developer needs [**?**]. In these studies, the bug reports have been extracted manually due to the unavailability of more efficient extraction methods, as well as human interpretation was required to extract selected parts of the bug report.

Observation 2.3 *Bug reports* present various challenges to researchers, mainly unavailability of more efficient extraction method to retrieve data and need of human interpretation to select relevant textual facts from it. Nevertheless, bug reports can be a valuable data source for future research as they often are connected to source code artifacts and commits, which can be used to link the communication with specific parts of the source code.

Newsgroups & Forums. Newsgroups or developer forums are similar to mailing lists in terms of purpose, community, structure, and level of technicality. Developers (or other interested persons) can discuss questions specific to a particular software project or a component. Hou *et al.* state that newsgroups can be valuable to debug problems, discussing issues or learning about existing bugs [**?**]. Two relevant studies concerning developer newsgroups [**?**], [**?**] extracted data from it manually and performed a manual analysis to investigate developer challenges with a specific software component. However, the research using newsgroup data is rather old and did not get subsequent attention. Whether researchers shifted away from using newsgroups as a data source or started to analyze different aspects require a thorough study.

User Reviews. User reviews are a relatively new source of information for researchers since app stores became only available with arrival of smart phones. In terms of structure, community, and level of technicality, user reviews are rather different from the other discussed sources. A user review consists only of text, potentially combined with a rating. The data is ideally suited to provide developers with more in-depth

insight into the user sentiments and opinions [**?**]. Based on this data source, researchers have proposed multiple solutions to guide release planning and evolution [**?**], [**?**], [**?**], or provided meaningful classification and clusters of user reviews [**?**], [**?**], [**?**]. However, the lack of an efficient extraction method is a significant drawback of this data source. The most popular app marketplaces do not provide an API to easily access and extract user review data. Thus, most of the discussed studies use web scraping as the extraction method for their data. This extraction method is not only a time-intensive task but also hard to reproduce and it suffers from a sampling bias which can ultimately affect the results [**?**].

Observation 2.4 *User reviews* are a more recent source for developers and researchers and help them in gathering opinions about the apps. User reviews studies lack efficient extraction methods, and suffer from the sampling bias thus making them hard to reproduce.

I. Tools comparison

TABLE II
TOOLS COMPARISON

Tool	URL	Costs	Data Mining	Preprocessing	Data Mgmt.	NLP	Visualization
Octoparse	[3]	Free (Limited)	Yes	No	No	No	No
Knime	[4]	Free	Extension	Yes	No	Yes	Yes
Pentaho Data Integration	[5]	Paid	No	Yes	?	No	?
Apache Mahout	[6]	Free	No	No	No	No	No
Birt	[7]	Free	No	No	No	No	Yes
Rapidminer	[8]	Commercial, Educational License	No	Yes	No	No	Yes
ELKI	[9]	Free	No	No	No	No	Yes
Keel	[10]	Free	No	No	No	No	Yes
WEKA	[11]	Free	Yes	Yes	No	No	Yes
MAKAR	[12]	Free	Yes	Yes	Yes	Limited	No

Title	Year	Reference
Information needs in bug reports: improving cooperation between developers and users	2010	[?]
An empirically-based characterization and quantification of information seeking through mailing lists during Open Source developers' software evolution	2015	[?]
Developing Schema for Open Source Programmers' Information-Seeking	2008	[?]
Open Source Programmers' Information Seeking During Software Maintenance	2011	[?]
How the R Community Creates and Curates Knowledge: A Comparative Study of Stack Overflow and Mailing Lists	2016	[?]
What Can Programmer Questions Tell Us About Frameworks?	2005	[?]
Empirical Analysis of the Logging Questions on the StackOverflow Website	2018	[?]
What are mobile developers asking about? A large scale study using stack overflow	2016	[?]
What are developers talking about? An analysis of topics and trends in Stack Overflow	2014	[?]
What Do Concurrency Developers Ask About? A Large-scale Study Using Stack Overflow	2018	[?]
How do programmers ask and answer questions on the web?: Nier track	2011	[?]
A Manual Categorization of Android App Development Issues on Stack Overflow	2014	[?]
What Concerns Do Client Developers Have When Using Web APIs? An Empirical Study of Developer Forums and Stack Overflow	2016	[?]
Towards comprehending the non-functional requirements through Developers' eyes: An exploration of Stack Overflow using topic analysis	2017	[?]
What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts	2016	[?]
Mining Questions about Software Energy Consumption	2014	[?]
Mining Questions Asked by Web Developers	2014	[?]
An Exploratory Analysis of Mobile Development Issues using Stack Overflow	2013	[?]
A Study on the Most Popular Questions About Concurrent Programming	2015	[?]
What Questions Do Programmers Ask About Configuration as Code?	2018	[?]
An empirical study on developer interactions in StackOverflow	2013	[?]
What are Software Engineers asking about Android Testing on Stack Overflow?	2017	[?]
Mining Testing Questions on Stack Overflow	2016	[?]
Obstacles in Using Frameworks and APIs: An Exploratory Study of Programmers' Newsgroup Discussions	2011	[?]
Development Emails Content Analyzer: Intention Mining in Developer Discussions	2015	[?]
Release Planning of Mobile Apps Based on User Reviews	2016	[?]
Crowdsourcing user reviews to support the evolution of mobile apps	2018	[?]
Why people hate your app: making sense of user feedback in a mobile app store	2013	[?]
Causal Impact Analysis Applied to App Releases in Google Play and Windows Phone Store	2015	[?]
The app sampling problem for app store mining	2015	[?]
AR-miner: mining informative reviews for developers from mobile app marketplace	2014	[?]
Mining User Opinions in Mobile App Reviews:A Keyword-based Approach	2015	[?]
CODES: Mining source code descriptions from developer communications	2012	[?]
User feedback in the appstore: An empirical study	2013	[?]
Mining Query Subtopics from Questions in Community Question Answering	2015	[?]
Attentive Interactive Convolutional Matching for Community Question Answering in Social Multimedia	2018	[?]
StackOverflow and GitHub: Associations Between Software Development and Crowdsourced Knowledge	2013	[?]
Using and Asking: APIs Used in the Android Market and Asked About in Stack Overflow	2013	[?]
Which Non-functional Requirements do Developers Focus on?	2015	[?]
Detecting API Usage Obstacles: A Study of iOS and Android Developer Questions	2013	[?]
Classifying Stack Overflow Posts on API Issues	2018	[?]