Information and Software Technology 57 (2015) 77–94

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# An empirically-based characterization and quantification of information seeking through mailing lists during Open Source developers' software evolution

Khaironi Y. Sharif [a], Michael English [b,*], Nour Ali [c], Chris Exton [b], J.J. Collins [b], Jim Buckley [b]

[a] Software Engineering Research Group (SERG), FSKTM, Universiti Putra Malaysia, Malaysia
[b] The Irish Software Engineering Research Centre (LERO), University of Limerick, Ireland
[c] School of Computing, Engineering and Mathematics, University of Brighton, UK

ARTICLE INFO

*Article history:*
Received 7 February 2014
Received in revised form 5 September 2014
Accepted 6 September 2014
Available online 16 September 2014

*Keywords:*
Information seeking software maintenance
Open source software
Qualitative empirical study

ABSTRACT

*Context:* Several authors have proposed information seeking as an appropriate perspective for studying software evolution. Empirical evidence in this area suggests that substantial time delays can accrue, due to the unavailability of required information, particularly when this information must travel across geographically distributed sites.
*Objective:* As a first step in addressing the time delays that can occur in information seeking for distributed Open Source (OS) programmers during software evolution, this research characterizes the information seeking of OS developers through their mailing lists.
*Method:* A longitudinal study that analyses 17 years of developer mailing list activity in total, over 6 different OS projects is performed, identifying the prevalent information types sought by developers, from a qualitative, grounded analysis of this data. Quantitative analysis of the number-of-responses and response time-lag is also performed.
*Results:* The analysis shows that Open Source developers are particularly implementation centric and team focused in their use of mailing lists, mirroring similar findings that have been reported in the literature. However novel findings include the suggestion that OS developers often require support regarding the technology they use during development, that they refer to documentation fairly frequently and that they seek implementation-oriented specifics based on system design principles that they anticipate in advance. In addition, response analysis suggests a large variability in the response rates for different types of questions, and particularly that participants have difficulty ascertaining information on other developer's activities.
*Conclusion:* The findings provide insights for those interested in supporting the information needs of OS developer communities: They suggest that the tools and techniques developed in support of co-located developers should be largely mirrored for these communities: that they should be implementation centric, and directed at illustrating "how" the system achieves its functional goals and states. Likewise they should be directed at determining the reason for system bugs: a type of question frequently posed by OS developers but less frequently responded to.



## 1. Introduction and motivation

Software maintenance and evolution are large components of a successful software system's lifecycle. The amount of software lifecycle effort consumed during this phase has been estimated to range between 60% and 80% of the entire lifecycle effort [1–4]. While the empirical basis for such statements is dated and suggestions have been made that it should be revisited [4], the increasing scale and complexity of newer software systems [3,5] implies that the effort invested in maintenance of successful systems can only have increased. Thus, research is required towards the discovery and evolution of supportive approaches or tools, which could improve efficiency in this effort-intensive activity.

Software maintenance can be divided into 2 general stages: "Understanding the program and actually performing the change" [6,7]. The time invested by a developer in order to achieve an understanding before (and during) a successful modification can be considerable, with typical estimates ranging from between 50% to 90% of the entire maintenance effort [8]. Information seeking has been recognized as a core subtask in software

* Corresponding author.

http://dx.doi.org/10.1016/j.infsof.2014.09.003
0950-5849/

comprehension within software maintenance [9–14]. Sim [13] for example, explicitly makes the link between software maintenance and information seeking, referring to maintenance programmers as "task-oriented information seekers", focusing specifically on getting the answers they need to complete a task using a variety of information sources.

Kingrey [15] defines information seeking as the searching for, recognition, retrieval and application of meaningful content. It is a difficult task in software evolution with developers often confused as to what to search for [16] and how to search for it [17]. In certain circumstances the information may not be readily available to them and this may stop their progress, an issue referred to as 'information blocking' [18]. While Ko et al. [19] states that information blocking is not 'inherently unproductive', empirical evidence suggests that, in practice, it is. Liu et al. [20], for example, find that information blocking can consume up to 60% of a developers time. Likewise Mockus et al.'s [21] study shows that information blocking could delay tasks by up to 0.9 of a day when the information is available in the same geographical location and by up to 2.4 days when the information has to come from a different geographical location.

The nature of Open Source (OS) software development makes it as an important context in which to study the difficulty of information acquisition during software maintenance and evolution. OS software is becoming increasingly important in its own right: many OS software systems are considered critical components of today's software landscape [22,23] and the OS community produce many long-lived systems that wide populations depend upon [24–26] over time.

Yet OS software development generally involves (or has the potential to involve) large, globally distributed communities of developers collaborating primarily through the internet [27,28] under differing governance models. As stated above, this typically widely-distributed, and asynchronous development environment would seem to make information seeking more difficult [27,29,30]. Thus information blocking in OS contexts is likely exacerbated when compared to co-located proprietary development [19,31].

This work focuses on the issue of information seeking for developers evolving Open Source projects. Specifically, it studies information seeking in 6 OS developer communities, as defined by developers' communication in their mailing lists, over 17 years, in total. It should be noted that mailing lists are not the only communication channels open to OS developers. In different OS governance models alternative channels may also be employed to facilitate communication, such as instant messaging applications, email, and/or face-to-face communication. These governance models and practices are discussed in more detail in Section 2.1, and the OS projects used in this study are characterized in terms of these governance models in Section 3.3.1

Analysis of alternative communication channels like email or instant messaging is outside of the scope of this paper. In addition this work does not consider solitary information seeking where the programmer refers to the code or documentation to seek the information they need. However there is an acceptance in the literature that mailing lists are the predominant communication channel of distributed OS development teams [30,21,32] and so can be informative in this regard. This is best illustrated with a quotation from the OpenOffice community [33].

"*Mailing lists are the backbone of open source communications*".

As a result, this research considers mailing lists as strongly representative of Open Source Developers' communication and argues that studying this communication channel can provide important insights on their Information Seeking needs. It identifies the information types they seek through this medium, focusing on their prevalently expressed information needs, post deployment. Subsequently it identifies the information types that seem difficult to acquire through this medium. It provides 3 specific contributions:

- A schema of information types sought by Open Source programmers through their mailing lists as they evolve software systems.
- Identification of the prevalent information needs across these OS communities, as expressed in their mailing lists.
- A determination of the response rates for different types of these information requests;

Identifying these prevalent question types allows researchers a framework in which to discuss response rate and quality. Analyzing the response rates with respect to these question types demonstrates the question types that the community are more (or less) willing to answer, thereby generating suggestions for information and tool support for these communities.

The paper is structured as follow: Section 2 discusses the related information-seeking work, contextualizing how the work reported here augments the existing body of work in the area. Section 3 describes the derivation of the schema used in the characterization of OS developers' mailing list questions. Section 4 reports on the resultant schema, the prevalent information request-types and the response rate of the community. Section 5 discusses these results and Section 6, concludes the paper.

It should be noted that this research is an extension of the work reported by the first and last authors at the Psychology of Programmers Interest Group (PPIG) [29,34]. The work reported on at PPIG describes a more preliminary version of the schema, based on a much smaller data-set (364 questions). This data-set did not cover several core categories of Open Source Software Systems, as reported in taxonomies such as Feller and Fitzgerald [27] and Daniel et al. [35]. The preliminary schema derived was then used to analyze that smaller data-set for prevalent information types and their associated response rates.

Here a more refined schema is presented, based on a larger, more representative data-set (708 questions) and more iterations of data analysis. This refined schema was used in this research to analyze the complete data-set giving a stronger empirical basis for refined insights regarding developers' information needs. A more detailed literature review is also included in this paper for comparison against these new findings.

## 2. Related work

### 2.1. OS governance

Information seeking in OS projects is contextualized by the governance model and working practices employed in those projects. The work presented in this paper focuses specifically on one aspect of governance, that is the use of information and tools [36]. This occurs in the context of two other aspects, these being the software development processes, and community management that implicitly guides these processes.

The two main types of governance models are where a single individual governs a project (known as a benevolent dictatorship) or alternatively a model that is governed by a group where control and membership of the group is regulated and often determined by the contributions of individuals to the project (known as a meritocracy) [37]. In the former model "seniority prevails" and individual contributors can become "responsible for one or more parts of the software" [38].