

# **An empirically-based characterization and quantification of information seeking through mailing lists during open source developers software evolution**

## **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.

**Keyword:** Information seeking software maintenance; Open source software; Qualitative empirical study