

Empirically Refining a Model of Programmers' Information-Seeking Behavior during Software Maintenance

Jim Buckley¹, Michael P. O'Brien¹, Norah Power¹

¹ Department of Computer Science,
University of Limerick, Limerick, Ireland

{Jim.Buckley, MichaelP.O'Brien, Norah.Power}@ul.ie

Abstract. Several authors have proposed information seeking as an appropriate perspective for studying software maintenance activities. However, there is little research in the literature describing holistic information-seeking models in this context. Additionally, in the one instance where an information-seeking model has been proposed, the empirical evidence presented in support of that model is extremely limited. This paper presents a small quasi-experiment that serves to further evaluate and refine this preliminary information-seeking model. Talk-aloud data, generated by two professional programmers, engaged in real software maintenance activities, was captured and then coded. This evaluation largely validated the model but also suggested several important refinements. The study, its results and its impact on the information-seeking model are discussed in this paper.

2 Introduction

Information seeking has been defined as the searching, recognition, retrieval and application of meaningful content [17]. Several researchers have argued that information seeking is a core element of software maintenance [8] [29], [31], [32]. Sim [32], for example, refers 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.

Seaman [29] and Singer [31] used questionnaire and interview-based empirical studies to further probe the information sources used by professional programmers during maintenance, and the factors that affected the perceived quality of these sources. They found that programmers relied predominantly on source code, a finding in agreement with that of [33]. However, other valued sources of information that these programmers identified were customers, users, the original development team, other system maintainers, 'Lessons-learnt' reports and execution traces (to recreate software bugs).

Bradac et al [4], and Liu et al. [22] have carried out some related research in the area of information 'Blocking'. Blocking arises when progress on a software engineering activity is halted because the engineers cannot get access to the information

