

The Effect of Task and Tool Experience on Maintenance CASE Tool Usage

Mark T. Dishaw, University of Wisconsin, USA
Diane M. Strong, Worcester Polytechnic Institute, USA

ABSTRACT

Computer-aided software engineering (CASE) tools have been advocated for improving maintainer productivity and the quality of maintained software. While there is evidence that such benefits can accrue to organizations adopting maintenance-oriented CASE tools, a key problem in achieving the desired benefits from CASE tools is low usage of these tools by programmers. The previously tested Maintenance Tool Utilization Model was a first step in investigating the factors that affect whether maintainers choose to use CASE tools during maintenance projects. We test the addition of experience with software maintenance tools and with the software maintenance task to the Maintenance Tool Utilization Model. The role of experience is important because managers can provide training to increase experience and they can ensure that project teams have some members experienced with the tools or with the task. Data for the test are collected from software maintainers working on their organization's normal maintenance project backlog. Tool experience is significant as both a main and interaction effect, but task experience adds little to the explanatory power of the Maintenance Tool Utilization Model. These results support the value of improved CASE tool training programs.

Keywords: Baseline Maintenance Tool Utilization Model Computer-Aided Software Engineering (CASE), Maintenance Tool Utilization Model

INTRODUCTION

Organizations are adopting Computer-Aided Software Engineering (CASE) tools to support maintenance tasks. Software maintenance is the process of changing existing, production software. Production software is changed to correct problems, to adapt the system to a changing hardware and software environment, and to improve the system by making it more efficient or by adding functionality for users (Swanson & Beath, 1989). Maintenance-oriented CASE tools provide soft-

ware tool support for the maintenance process. Such tools have been identified as a key to achieving maintenance productivity gains (Schneidewind, 1987). In addition to improving productivity, the use of these tools may contribute significantly to improving the quality of the software being maintained (Kim & Westin, 1988).

If CASE tools are used, productivity and quality benefits are being achieved (see Iivari, 1996, for a brief review of this literature). The problem, however, is low utilization of these tools (Iivari, 1996; Kemerer, 1992). Since CASE tools that

are not used will have no effect on maintainer productivity or software quality, positive or negative, some amount of utilization is required for benefits to accrue to the organization. While some argue that too much utilization does not provide additional benefits, few would argue that overuse of software maintenance tools is currently a problem. Low tool utilization is the practical problem organizations adopting CASE tools now face. Tool utilization is the outcome measure used in this study.

The software maintenance process consists of two major steps, understanding the existing production system and modification of this software (Pennington & Grabowski, 1990; Yau & Collofello, 1985). The software maintenance support tools of interest to us are generally intended to support program understanding. Such tools are of interest because maintenance programmers spend 50-90% of their time understanding the program (Shaft & Vessey, 1998). If tools could help in the understanding process, the potential for significant productivity enhancements is large. These tools assist the programmer/analyst in discovering the physical and logical designs of the program or system at hand. The discovery of impacts of a proposed change on "distant" programs, i.e., programs linked by common data elements, is an important part of this phase, and is assisted by these software tools.

Software maintenance support tools may also contain functionality that facilitates the coordination of programmers on a large project. Such functionality is provided by a variety of schedule and project management programs, and includes communications programs such as e-mail, and audio and video conferencing.

This research study extends previous work that developed a Maintenance Tool Utilization Model (Dishaw & Strong,

1998b). The Maintenance Tool Utilization Model is based on the task-technology fit (TTF) literature (e.g., Goodhue, 1995; Goodhue & Thompson, 1995). The argument of TTF-based models is that software tools will be used if they fit the needs of the task. The utilization of maintenance-oriented CASE tools can be explained by TTF-based models (Dishaw & Strong, 1998a, 1998b). Fit between maintenance tool functionality and the needs of the maintenance task, however, is only part of the reason a maintainer chooses to use, or not to use, maintenance tools for a particular maintenance project. Other variables are likely to affect a maintainer's decision to use a tool.

Of particular interest is the role of experience with CASE tools in explaining tool utilization (Guinan, Coopride, & Sawyer, 1997; Thompson, Higgins, & Howell, 1994; Venkatesh & Davis, 1994, 1996). When purchasing tools, managers may fail to invest sufficiently in training for potential tool users. If tool experience is critical to utilization decisions, it has implications for how management implements maintenance-oriented CASE tools. In particular, training may be an important component of successful CASE tool adoption.

In the MIS literature, task experience is also considered to be an important variable (Mackay & Elam, 1992; Shaft & Vessey, 1995). If task experience is critical for utilization decisions, management must consider the maintenance experience profile of their maintenance staff when making decisions on purchasing maintenance-oriented CASE tools.

In this study, the Maintenance Tool Utilization Model is extended to include tool and task experience variables. The effect of these additional variables on maintenance tool utilization is tested to determine their contribution to utilization beyond that

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/effect-task-tool-experience-maintenance/1245

Related Content

Understanding the "Mommy Tracks" : A Framework for Analyzing Work-Family Balance in the IT Workforce

Jeria L. Quesenberry, Eileen M. Trauthand Allison J. Morgan (2006). *Information Resources Management Journal* (pp. 37-53).

www.irma-international.org/article/understanding-mommy-tracks/1290

Let's Get United and #ClearTheShelters: The Factors Contributing to Users' Network Centrality in Online Social Networks

Ezgi Akar (2022). *Journal of Information Technology Research* (pp. 1-18).

www.irma-international.org/article/lets-get-united-and-cleartheshelters/299943

Information Management: Jurisdictional, Legal and Ethical Factors

Michael Losavio, Adel Elmaghrabyand Deborah Keeling (2009). *Open Information Management: Applications of Interconnectivity and Collaboration* (pp. 406-420).

www.irma-international.org/chapter/information-management-jurisdictional-legal-ethical/27806

An Empirical Study of the Factors that Influence In-Class Digital Distraction among University Students: A U.S. – Namibia Cross-Cultural Study

Ravi Nath, Leida Chenand H. N. Muyingi (2015). *Information Resources Management Journal* (pp. 1-18).

www.irma-international.org/article/an-empirical-study-of-the-factors-that-influence-in-class-digital-distraction-among-university-students/132764

Leveraging Knowledge Reuse and Systems Agility in the Outsourcing Era

Igor Crk, Dane Sorensenand Amit Mitra (2008). *Journal of Information Technology Research* (pp. 1-20).

www.irma-international.org/article/leveraging-knowledge-reuse-systems-agility/3694