



Chapter VIII

Validating an Evaluation Framework for Requirements Engineering Tools

Raimundas Matulevičius,
Norwegian University of Science and Technology, Norway

ABSTRACT

Automated support for the requirements engineering (RE) process is a recognized research area. However, the mainstream practice still relies on word processors and drawing tools rather than the requirements engineering tools (RETs). The aim of this chapter is to validate an evaluation framework for RETs. The validation process concerns an RET acquisition process for concrete organizational needs. An observation of maintaining requirements specification shows the important organizational and environmental characteristics for a proper automated support of RE process. The contribution of this work is twofold: first, the validation of the evaluation framework for RETs according to environmental needs in a specific environment, and second the identification of environmental needs, which emerge from the requirements specification maintenance process.

INTRODUCTION

The usefulness of automated support for the software development process is recognized in the literature (Kotonya & Sommerville, 1998; Harrison, Ossher, & Tarr, 2000; Nuseibeh & Easterbrook, 2000; Kaindl et al., 2002), but the mainstream of requirements engineering practice still relies on word processors and drawing tools, rather than targeted tools provided by various researchers and practitioners. Current commercial off-the-shelf (COTS) tools for requirements engineering provide capabilities for documenting requirements and facilitating requirements management. As considered in Kaindl et al. (2002) and Karlsson et al. (2002), the tools are well suited for managing large amounts of requirements written in natural language, but not for engineering the requirements. Requirements engineering tools are usually described as CASE (computer-aided system engineering) tools. CASE technologies are defined as tools that provide automated assistance for software development (Lending & Chervany, 1998). In the early 1980s requirements engineering seemed to be a relatively simple task, and existing CASE tools were expected to provide task-related support for software developers. But as discussed in Kelly, Lyytinen, and Rossi (1996), the weaknesses of CASE tools had become apparent, so product and process quality improvement by using CASE tools remains questionable. Kaindl et al. argue that one plausible reason for this is the lack of maturity to adopt tools. Kelly et al. stress the apparent cost of adopting, using, and maintaining a tool, and the inadequate technological sophistication of the CASE tools. Fitting the requirements engineering tools to meet customer requirements remains problematic because companies employ different software analysis, modeling, and engineering methods. Requirements engineering tools vary in their level of support for requirements engineering activities. The evaluation for selection purposes has to be performed before buying any tools. A company cannot base evaluation on its own long-term tool use. Instead, it can only rely on tool surveys, commercial reports, which are unreliable because in many cases they depend on vendor information and become quickly out of date.

Evaluation of requirements engineering tools differs depending on the environment, needs, and purposes for tool usage. Botella et al. (2002) investigate two questions during the evaluation of available software tools. First, how are the tools of a given domain described in order to make their comparison feasible? Second, how may the features of the tool be reconciled with respect to requirements for tools? Evaluation and acquisition of a requirements engineering tool to organizational needs should fall into the feasibility limits of an organization. Requirements engineering tools could be evaluated from a theoretical point of view, practical experience from industry could be gathered and evaluated, or tools could be tried out on some realistic examples under organizational settings. The evaluation and comparison would be more complete and structured if an evaluation framework, which targets the questions, is applied. A framework is

25 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/chapter/validating-evaluation-framework-requirements-engineering/23013

Related Content

Issues in Transaction-Time Temporal Object Database Systems

Kjetil Norvag (2001). *Journal of Database Management* (pp. 40-51).

www.irma-international.org/article/issues-transaction-time-temporal-object/3271

Applying JAVA-Triggers for X-Link Management in the Industrial Framework

Abraham Alvarez and Y. Amghar (2003). *Effective Databases for Text & Document Management* (pp. 135-154).

www.irma-international.org/chapter/applying-java-triggers-link-management/9209

Methodology of Schema Integration for New Database Applications: A Practitioner's Approach

Joseph Fong, Kamalakara Karlapalem, Qing Li and Irene Kwan (1999). *Journal of Database Management* (pp. 2-18).

www.irma-international.org/article/methodology-schema-integration-new-database/51209

Replication Mechanisms Over a Set of Distributed UDDI Registries

Zakaria Maamar (2005). *Encyclopedia of Database Technologies and Applications* (pp. 548-554).

www.irma-international.org/chapter/replication-mechanisms-over-set-distributed/11203

A Case Study of the Military Utility of Telemedicine

David J. Paper, James A. Rodger and Parag C. Pendharkar (2001). *Developing Quality Complex Database Systems: Practices, Techniques and Technologies* (pp. 350-365).

www.irma-international.org/chapter/case-study-military-utility-telemedicine/8284