Chapter 2.20
Riki:
A System for Knowledge Transfer and Reuse in Software Engineering Projects

Jörg Rech
Fraunhofer Institute for Experimental Software Engineering (IESE), Germany

Eric Ras
Fraunhofer Institute for Experimental Software Engineering (IESE), Germany

Björn Decker
Fraunhofer Institute for Experimental Software Engineering (IESE), Germany

ABSTRACT

Many software organizations have a reputation for producing expensive, low-quality software systems. This results from the inherent complexity of software itself as well as the chaotic organization of developers building these systems. Therefore, we set a stage for software development based on social software for knowledge and learning management to support reuse in software engineering as well as knowledge sharing in and between projects. In the RISE (Reuse in Software Engineering) project, we worked with several German SMEs to develop a system for the reuse of software engineering products such as requirement documents. The methodology and technology developed in the RISE project makes it possible to share knowledge in the form of software artifacts, experiences, or best practices based on pedagogic approaches. This chapter gives an overview of the reuse of knowledge and so-called Learning Components in software engineering projects and raises several requirements one should keep in mind when building such systems to support knowledge transfer and reuse.

INTRODUCTION

The software industry develops complex systems that often have a reputation of being expensive and of low quality. One approach for coping with the increasing complexity of these systems is software reuse—the sharing of knowledge about software products, processes, people, and projects in an organization.

But the poor and often nonexistent documentation of this knowledge inhibits easy recording, collection, management, comprehension, and transfer. The knowledge, for example, in the form of requirement descriptions, architectural information, design decisions, or debugging experiences, needs a systematic, minimally invasive, methodological, and technological basis to strengthen its reuse and transfer in software organizations.

Knowledge management (KM) and *learning management* (LM) seem to have the potential for building this basis if they are used in synergy. However, the relationships between these two promising fields have not been fully understood and harnessed yet. On the one hand, learning as the comprehension of knowledge is considered to be a fundamental part of KM, as employees must internalize shared knowledge before they can use it to perform a specific task. So far, research within KM has addressed learning mostly as part of knowledge sharing processes, and focused on specific forms of informal learning (e.g., learning in a community of practice) or on providing access to learning resources or experts. On the other hand, LM includes techniques to preprocess and formalize knowledge, and might also benefit from other KM approaches. Especially, those approaches that support technical and organizational aspects in an organization can be used in addition to professional e-learning systems.

In this intersection between KM and LM, the Wiki technology (cf. http://en.wikipedia.org/wiki/Wiki) promises to be a lightweight basis to capture, organize, and distribute knowledge that is produced and used in organizations. Wikis are Web-based knowledge repositories where every user can freely submit or retrieve knowledge.

The *RISE* (*Reuse In Software Engineering*) project was started to support software engineers via the reuse of didactically enriched software artifacts from all software development phases in SMEs (small and medium enterprises). As the basis for knowledge transfer, we have developed a reuse-oriented Wiki (Riki) with fast and liberal access to deposit, mature, and reuse experiences made in software projects. Our Riki was extended by search technologies using case-based reasoning technology and ontologies to provide a formal and consistent framework for the description of knowledge and experiences. Ontologies and templates enrich the Riki content with semantics that

Figure 1. Outline and structure of the following chapter