Chapter XVII
Knowledge Patterns and Knowledge Refactorings for Increasing the Quality of Knowledge

Jörg Rech
Fraunhofer Institute for Experimental Software Engineering, Germany

Raimund L. Feldmann
Fraunhofer USA, Center for Experimental Software Engineering, USA

Eric Ras
Fraunhofer Institute for Experimental Software Engineering, Germany

Andreas Jedlitschka
Fraunhofer Institute for Experimental Software Engineering, Germany

Björn Decker
empolis GmbH, Germany

ABSTRACT

Knowledge management is a relatively young discipline. It has accumulated a valuable body-of-knowledge on how to structure and represent knowledge, or how to design socio-technical knowledge management systems. A wide variety of approaches and systems exist that are often not interoperable, and hence, prevent an easy exchange of the gathered knowledge. Industry standards, which have been accepted and are in widespread use are missing, as well as general concepts to describe common, recurring patterns of how to describe, structure, interrelate, group, or manage knowledge elements. In this chapter, we
Knowledge Patterns and Knowledge Refactorings for Increasing the Quality of Knowledge

The concepts “knowledge pattern” and “knowledge anti-pattern” are introduced to describe best and worst practices in knowledge management, “knowledge refactoring” to improve or change knowledge anti-patterns, and “quality of knowledge” to describe desirable characteristics of knowledge in knowledge management systems. The concepts are transferred from software engineering to the field of knowledge management based on our experience from several knowledge management projects.

INTRODUCTION

The ability to learn from knowledge gained in past projects to determine success factors and reasons for failure is a key factor for organizational learning. Consequently, knowledge management (KM) as well as learning management (LM) are important for any quality improvement initiative to succeed. Researchers and practitioners have defined and installed a multitude of models, theories, and systems comprised of valuable and recurring knowledge that is waiting to be reused in KM systems. However, the quality of the knowledge gained, the technical KM system used, or the social KM method applied is neither easy to evaluate, nor is it easy to improve. This is partly due to the fact that there exists no universal KM system that is suitable for all kinds of organizations. In practice, each system has to be adapted and tailored to the individual needs of an organization. Examples of aspects influencing the design of KM systems include the improvement and learning strategies applied, the physical structure of the organization (e.g., different sites and locations), and the organization’s logical structure (e.g., departments, projects, and working groups). Defining a KM initiative and implementing a KM system in an organization remains a time consuming and often error-prone endeavor. As a result, knowledge on how to avoid errors or shorten the time for implementation is in high demand.

Such knowledge on KM systems has been documented in the form of success factors (Mathi, 2004) (Thomas, 2006) (Morisio et al., 2002), success models (Jennex & Olfman, 2004, 2006), success measures (Jen & Yu, 2006), reference architectures for KM systems (Davenport & Probst, 2000; Mertins, 2003), or worst practices (Fahey & Prusak, 1998). They typically preserve knowledge about a whole KM system or initiative. But (complete) reuse of existing solutions is neither common nor do standard schema libraries or COTS collections for creating a KM system exist. Hence, support for a consistent set of KM systems, which would allow for easy integration of existing knowledge into an organization’s specific system, or for sharing and exchanging knowledge with other organizations, is still missing. Especially such easy exchange and wide-spread use would help to detect flaws and misbelieves in the existing body of knowledge, and thereby would help to significantly increase the quality of knowledge. Even commonly accepted best practices on how to structure knowledge, how to design an interface for a KM system, or how to start a storytelling session are hard to find.

Unlike in other disciplines, general concepts or rule of thumbs documenting recurring patterns of how to describe, structure, interrelate, group, or manage knowledge are still missing.

During the mid-1990s, the concept of “design patterns” was developed in software engineering to describe best practices regarding the design of software systems in a structured way. Design patterns are used to represent knowledge that is based on experiences captured in several real-world projects and is widely accepted. This semi-formal representation is often used for describing and presenting the gained knowledge.

In this chapter, we transfer the concepts of quality, patterns, and refactoring from software engineering to the field of KM and introduce the concepts of knowledge patterns and knowledge...