Chapter 2.19
Human-Centered Design of a Semantically Enabled Knowledge Management System for Agile Software Engineering

Christian Höcht
Technical University of Kaiserslautern, Germany

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

ABSTRACT

Developing human-engineered systems is considered as a challenge that addresses a wide area of expertise; computer scientists as well as social scientists. These experts have to work together closely in teams in order to build intelligent systems to support agile software development. The methodology developed in the RISE project enables and supports the design of human-centered knowledge-sharing platforms, such as Wikis based on standards in the field of education science. The project "RISE" (Reuse In Software Engineering) is part of the research program "Software Engineering 2006" funded by the German Federal Ministry for Education and Research (BMBF). The goal was to improve the reuse of artifacts in software engineering, and brought together researchers from education science (The Department of Educational Sciences and Professional Development at the Technical University of Kaiserslautern) and computer science (Fraunhofer Institute for Experimental Software Engineering (IESE) and the German Research Center for Artificial Intelligence (DFKI)) with industrial partners (Empolis GmbH and brainbot technologies AG). This chapter gives an overview about the human-centered design of Wiki-based

knowledge and learning management systems in software engineering projects, and raises several requirements one should keep in mind when building human-centered systems to support knowledge and learning management.

INTRODUCTION

The development of complex software systems is based on company- and domain- specific knowledge that has to be constantly cultivated among the employees, because the resulting quality of manufactured software systems depends on what degree the needed knowledge is actually available (Decker, Ras, Rech, Klein, Reuschling, Höcht, & Kilian, 2005). It is not only that the technical platform should release software engineers as much as possible from time-consuming retrieval-processes. At the same time the platform acquires valuable pieces of information from users, who publish their problems and experiences during work. Out of this process, company-based knowledge may be built and refined.

However, it is just because of the various possibilities of searching and browsing through artifacts that users feel overwhelmed by the flood of information. Therefore, the increasing amount of information itself is not the problem, but an unfiltered and unrated access to it. In fact, the main goal is to ensure that software engineers can deal with their daily tasks without burdening them with additional work. A systematic selection and presentation of content helps to avoid the feeling of being swamped with artifacts and so keeps, and even improves, the employees' motivation. For example, on the one hand, it is a good idea to offer plugins that provide visual representations of concepts, but on the other hand, offering totally unrestricted overviews is often too complex to be processed by the users (Tonkin, 2005).

Ballstaedt (1997) mentions four main types of artifacts considering textual content. As Wikis mainly consist of text, this typology is helpful to

Figure 1. Different types of content

	Description	Example	Challenges in Knowledge Management related to Software Engineering
expository text	Expository text describes facts, explains the context, and thus provides conceptual knowledge.	dictionary entry	Information related to software products, solutions, or methods might expire very quickly, and would thus require much effort to maintain it.
narrative text	Narrative text reports actions or plots and events. It informs about specific situations, motives, decisions, acts, and their consequences.	blog entry	Narrative text might be rather helpful for some employees, but probably just a waste of time for others, because the individual usefulness of a specific narrative text is hard to evaluate.
instructional text	Instructional text provides procedural knowledge and enables people to do or not to do something.	help on Installation or maintenance	Providing useful instructional text assumes that the author respects different levels of knowledge and writes adequate content, keeping a specific target group in mind.
additional didactic text	Additional text that supports specific learning activities.	advance organizer	Additional didactic content requires deeper knowledge about reading and learning strategies and depends on rather static content like printed material.