

# A Framework for Agile Reuse in Software Engineering using Wiki Technology

Björn Decker<sup>1</sup>, Eric Ras<sup>1</sup>, Jörg Rech<sup>1</sup>, Bertin Klein<sup>2</sup>, Christian Reuschling<sup>2</sup>,  
Christian Höcht<sup>3</sup>, Lars Kilian<sup>3</sup>, Ralph Traphoener<sup>4</sup>, and Volker Haas<sup>5</sup>

<sup>1</sup> Fraunhofer Institute for Experimental Software Engineering (IESE),  
67661 Kaiserslautern, Germany  
{bjoern.decker, eric.ras, joerg.rech}@iese.fraunhofer.de

<sup>2</sup> German Research Institute for Artificial Intelligence (DFKI),  
67663 Kaiserslautern, Germany  
{bertin.klein, christian.reuschling}@dfki.de

<sup>3</sup> Chair of Pedagogics, Technical University of Kaiserslautern  
67663 Kaiserslautern, Germany  
{hoecht, lkilian}@rhrk.uni-kl.de

<sup>4</sup> empolis GmbH  
67657 Kaiserslautern, Germany  
ralph.traphoener@empolis.de

<sup>5</sup> brainbot technologies AG  
55118 Mainz, Germany  
volker.haas@brainbot.com

**Abstract.** The development of software is often accompanied by poor documentation of experiences, decisions, or architectural information. Reuse of experiences and knowledge, using the lightweight Wiki-Technology, promises a sustainable assurance and availability of business-critical information. In this paper we give an overview of the RISE research project (Reuse In Software Engineering), state the targeted research issues, and present our approach to tackle several of the identified problems.

## Introduction

The methodic reuse of software artifacts was first motivated by Dough McIllroy on the 1968 NATO conference of Software Engineering. Despite this long tradition, systematic reuse is still facing several challenges. These challenges are caused by insufficient support for the reuse steps [2]: *Search (and Retrieval)*, *Evaluation*, *Adaptation*. Concerning *search*, people do not find existing artifacts or do not even start to search due to the effort related with it. *Evaluation* challenges are mostly caused by either lengthy or insufficient documentation of the artifact found. This leads often to bad understanding and thus difficulties in evaluating the retrieved artifacts. Finally, *adaptation* refers to the (perceived) effort to understand and adapt the artifact in contrast to the effort to its first creation.

In order to realize reuse in an agile development environment Wikis [6] can be seen as a lightweight KM platform for exchanging reusable artifacts between and within software projects. From our perspective a Wiki system can be considered as lightweight *Organizational Memories* [3] or *Experience Factories* [4]. Wikis facilitate the communication by a basic set of features and delegate the actual way of coordination to the people using the Wiki. From the authors' point of view, these basic features are: *one place publishing* means, that there is only one version of a document available that is regarded as the current version; *simple and safe collaboration* refers to versioning and locking mechanisms that most Wikis provide; *easy linking* means that documents within a Wiki can be linked by their title using a simple markup; *description on demand* means that links can be defined to pages that are not created yet, but might be filled with content in the future.

The overall goal of the RISE project is to integrate lightweight experience management with agile software development. RISE faces the following goals:

- *Improvement of communication* between employees to strengthen and accelerate the transfer of knowledge face-to-face and via technical systems (e.g., synchronous chats or asynchronous forums).
- *Improvement of the retrieval of knowledge and orientation* in knowledge spaces to optimize the amount and accelerate the time to access relevant knowledge.
- *Improvement of the quality of knowledge* by assisting Software Engineers to create optimized artifacts (i.e., with optimized content and structure) based on didactical principles. These artifacts can easily be adopted and internalized by users of different expertise levels in order to support them in their daily work and to narrow competence gaps.
- *Improvement of utilization and usability* of technical KM systems to enable the user the goal-oriented search for suitable solutions to his problem in minimal time and to support him in the adaptation of the solution to his specific problems.

## **RISE: A Framework for Reuse in Agile Software Engineering**

The objective of the RISE project is the improvement of reuse in software organizations and focuses on the development of a minimal-invasive and user-centered system for the deposition, (re-)use, and advancement of information, experiences, and knowledge during agile software development. Our approach in the research project RISE is to develop new concepts on top of several approaches from knowledge management, artificial intelligence, pedagogics, and software engineering. The following methods and technologies will be analyzed, implemented, and evaluated:

- To improve the *knowledge preservation and exchange across project borders* we are using Wikis as a mean for asynchronous communication as well as an experience base. Furthermore, synchronous communication will be supported via chats to enable the fast exchange and rapid build-up of “communities of practice” (COP's) [5] on demand (i.e., accelerated discovery and communication with experts in a distributed organization). Information from these communication

platforms as well as data sources like CVS or defect tracking systems (e.g., Bugzilla) will be integrated into a knowledge map (e.g., by associating people with topics or subsystems) to improve the services to the user.

- The improvement of *knowledge retrieval* and *knowledge integration* will be tackled with ontology-based search engines, adaptive search masks, and integration of external sources. The ontology will be extracted from and synchronized with the metadata in the Wiki-pages. User- and task-specific search masks will be used to systematic select and filter content to improve the individual orientation. The external search will help to include information from open sources in the internet like defect tracking systems, newsgroups, FAQ's, or technology reports.
- We will enhance the *knowledge quality* and *knowledge mediation* through the augmentation with additional, explaining, and defining information in so called "micro-didactical experience arrangements". Furthermore, users will be supported in the process of recording experiences with didactical instruments or structural guidelines. This is targeted to unify the content structure and therefore improve the comprehension of artifacts from the repository.
- The problem of *insufficient usage* will be addressed by methods to foster the usage and operation of Wikis, i.e., systematic methods to improve the quality of the content as well as defining when and how to use the features of the Wiki. This is in particular valid for software engineering content that becomes quickly outdated or obsolete. Furthermore, like most technical support for communication among humans, the transfer of knowledge using technology like Wikis is typically inferior to face-to-face communication. Didactical useful settings seem to depend mainly on a strictly user-centered design but users have very different expectations based on their expertise.
- The problem of a *decaying structure* will be addressed by technical support that provides access to the content in addition to the link structure caused by manual linking of content.
- The problem that *relevant content from outside the Wiki is not regarded* will be tackled by integrating references to sources of knowledge from outside the Wiki, which then can be included by Easy Linking into the Wiki. Beside the obvious sources like websites or newsgroups in the internet there are also more software engineering specific sources like code repositories (e.g., CVS) or bug tracking systems (e.g., bugzilla).

Reuse in software engineering needs support in order to work in agile software organizations. Poor documentation and management of knowledge, experiences, decisions, or architectural information accompanies the development of software with agile methods in distributed software organizations. The Wiki-technology promises a lightweight solution to capture, organize, and distribute knowledge that emerges and is needed fast in agile software organizations.

The RISE framework sketches our approach for agile reuse and will tackle several problems in traditional KM and agile software organizations. Semi-automatic indexing of pages will improve the retrieval and will enable the semi-automatic creation and evolution of ontologies from Wikis (i.e., Wikitologies). The cooperative adaptation of knowledge to community needs, and the didactic augmentation of the

content and interface are targeted to improve the usability of lightweight KM applications in agile environments.

As a basis we are using Wikis as repositories with fast and liberal access to deposit, mature, and reuse experiences made in agile projects. Our next step is the design and implementation of additional functionality to a Wiki with a first version targeted for spring 2005. In the context of our project, we pursue the following research questions:

- *Are free structures of experience content and hierarchy more accepted by the users than fixed structures?* A long term goal would be dynamic or individual structures based on personal arrangement of documents to structural elements (e.g., folders) coded interpretation of metadata.
- *Does the extraction of information from existing sources (e.g., versioning, defect tracking) improve the integration of experiences and experts?* This will improve the access on experts and knowledge-carriers and will ease the build-up of goal-oriented face-to-face communication.
- *Does the augmentation of experiences improve the experience's understandability and applicability, compared to conventional, non-enriched experience descriptions?* A long term goal is to provide mechanism for creating micro-didactical experience arrangements by considering appropriate instructional design theories.
- *Is wiki-supported management of experience better accepted by the users than classical experience management applications in agile processes?* Classical experience management applications need a well defined process to be integrated. Wikis – in particular enhanced by ontologies – might provide a solution for agile and hence less structured processes.

## Acknowledgements

Our work is part of the project RISE (Reuse in Software Engineering), funded by the German Ministry of education and science (BMBF) grant number 01ISC13D.

## References

- [1] Mili, H., Mili, Ali, Yacoub S., Addy, E.: *Reuse-Based Software Engineering – Techniques, Organization and Measurement*, John Wiley and Sons (1995)
- [2] Karlsson, Even-Andre (Ed.): *Software Reuse – a holistic approach*, John Wiley and Sons (1995)
- [3] Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., and Sintek, M.: *Towards a Technology for Organizational Memories*. IEEE Intelligent Systems, 13(3), May/June 1998.
- [4] Basili V. R., Caldiera G., and Rombach H. D., *Experience Factory* in Encyclopedia of Software Engineering, vol. 1, J. J. Marciniak, Ed. New York: John Wiley & Sons, 2001, pp. 511-518.
- [5] Wenger, Etienne: *Communities of practice. Learning, meaning, and identity*. Cambridge University Press.
- [6] Leuf, Bo; Cunningham, Ward: *The Wiki Way*, Addison-Wesley Professional, (2001)