

Wikis for Collaborative Software Documentation

Claudia Müller

(University of Potsdam, Germany
cmueller@wi.uni-potsdam.de)

Lukas Birn

(PSIPENTA Software Systems GmbH, Germany
lbirn@psipenta.de)

Abstract: Software documentation is often a neglected aspect of software development although high quality end user documentation is an important competitive instrument. The process of creating such documentation is knowledge-intensive. In this contribution an end user documentation refers to a collection of knowledge which is compiled by various distinct knowledge sources within a company. To meet the highly specialised needs of such documentation a wiki is implemented. The background, the implementation procedure, and the characteristics of a corporate wiki are introduced.

Keywords: Software Documentation, Knowledge Collection, ERP, Social Software, Wiki

Categories: D.2.0, F.1.1, H.3.2, H.3.4, H.4.2, H.5.4, J.4, K.4.2, K.4.3

1 Knowledge Work in Software Development Companies

A company is an open system of social actions, which asserts itself in a turbulent environment through a high degree of self regulation [Emery 00]. Due to changing customer requirements, companies are subject to ongoing change processes [Wohlgemuth 91, p.37]. In this respect, software companies operate in a particularly turbulent environment. Software is created in collaborative, non-sequential, but parallel, interacting processes. To ensure the long-term success of a company, employees should not merely be treated as cost drivers or resources, but rather as essential contributing factors to its turnover. A company should implement target-oriented development and sustainable utilization of its people's competencies. Due to the increasing importance of knowledge, the "industrial paradigm", which dominated business in the past, is being replaced by the "knowledge paradigm" [Sveiby 97, p.28]. This is accompanied by an approach that a rather tayloristic (traditional) influenced picture of a company should be abandoned in favour of an organically structured, flexible and dynamic (agile) working method.

2 The Increasing Importance of Software Documentation

The process of creating end user documentation is usually neglected. Software developers typically consider end user documentation as a necessary evil, as creating it is often a very demanding task and the underlying benefit is difficult to measure. In large software companies an in-house "technical writing" department is often

responsible for the creation of end user documentation. Such documentation is a product of contributions from various knowledge domains within the company. For example software developers supply basic contents, a technical writer steers the content editing process and in special cases, the marketing department defines the format. The involvement of a number of people from various knowledge areas in this process – referred to as process participants - leads to a lack of clarity about their individual inputs. This means that process participants are partly unaware of their responsibility in the creation process. Moreover because documentation is created exclusively in one department, the result of the process is not visible for some within the company. Therefore it is also difficult to communicate to process participants the importance of their contributions to the product (documentation) as a whole. It must be taken into account that the quality of documentation has an effect on companies' success in the short-term as well as in the long-term. End user documentation is an important advertising medium. The short-term consequences of poor documentation quality include an increase in support requests, which then brings about additional costs. To avoid competitive disadvantages such costs cannot be allocated to customers, and as such the company's turnover decreases.

3 Case Study – Wiki based Software Documentation

In the introduced project the end user documentation development process of an ERP system was documented, analysed and redesigned based on identified potentials. In order to describe the complex processes of an ERP system completely but yet in an uncomplicated manner, the knowledge of diverse process participants must be interconnected. Therefore documentation is developed through close collaboration and intense interaction. The process of creating and constructing a new documentation version consumes a considerable amount of time because the documentation of an ERP system consists of thousands of so-called topics. The documentation grows in an iterative fashion during the development process. With every software release new functions are added or superfluous elements are removed, so that existing content has to be expanded or modified. The documentation's content is very interdependent. New functions must be interwoven into many different sections of the documentation. The more detailed the documentation, the more sections must be revised. This is even more important if the same topic is edited for different target groups (e.g. key users and 'standard' end users). Weak integration of the documentation and development tools can lead to high reluctance to carry out minor documentation modifications such as typo corrections. A frequent problem is that the documentation is related too close to peculiarities of the software internals, making it unnecessarily complex.

The process analysis was carried out using the KMDL®. The modelling method KMDL® (Knowledge Modeling and Description Language) serves as a basis for the modelling and analysis, then based on this, redesign of knowledge intensive business processes [Gronau et al. 04], [Müller et al. 05]. Based on captured KMDL® models, a process analysis was carried out. Amongst other objectives it was determined that efforts to change, extend, or update the end user documentation must be reduced (at the time of process analysis more than eight hours were required to generate an updated or changed documentation). To remove rather trivial errors (e.g. one

misplaced comma) the effort required was disproportionate to the result. Furthermore it was determined that the very high complexity of communication and coordination must be reduced.

A specified to-be concept covered organisation and technique areas, while motivational activities were considered within an implementation concept. Taking the determined potentials and the existing requirements such as combining the distributed knowledge resources, introducing an intuitional tool, and utilising a low capital expenditure, into account, it was decided to suggest the implementation of a wiki.

The first wiki was developed by Ward Cunningham as tool for knowledge management. A wiki is described as “a freely expandable collection of interlinked Web “pages”, a hypertext system for storing and modifying information – a database, where each page is easily editable by any user” [Leuf and Cunningham 01, p.14]. Wikis are perfectly suited for creating and managing a non-hierarchical text or knowledge collection because of the very easy principles of handling and linking single pages. There is no predetermined structure; instead all participants develop one in a self-organised process. Therefore, a wiki is not a passive information instrument, but rather an active one. Based on existing comparisons (cp. [WikiEngines 05], [WikiMatrix 06]), a decision was taken to employ MediaWiki. Existing functions of MediaWiki are customised to meet the special requirements.

The new solution for software documentation consists of three main components, of which the wiki is the core component. The wiki is used for data storage and for communication purposes. Utilisation of wiki templates ensures a consistent layout. Furthermore, templates allow for the potential to make global changes in a later development stage. The interface between the ERP system and end user documentation was raised to a higher level of flexibility. The so-called help keys within the source code were removed. The new approach utilised the redirect feature of MediaWiki. The ERP system references the documentation in its actual context (e.g., a special field or form). The wiki has a special page for every issue or context and redirects the user to corresponding topics. The knowledge for the appropriate documentation section is transferred from the ERP system to the documentation itself.

In order to generate new content, the PSIwikiRefresh component was developed. This tool is integrated into the development framework of the ERP system. The component compares the GUI (graphical user interface) elements with the documentation elements. If elements of the latter are missing, the corresponding items are generated based on predefined templates. The software developer generates the documentation while completing the software functionality. The newly generated documentation is transferred to the technical writer with the integrated communication mechanisms of the wiki.

The epos2wikiconverter is used to transform existing content. During the migration the original content is parsed and restructured. An important aspect is to ensure that the content linking is not broken during transformation and that it functions fully in the wiki. A high degree of automation and minimal manual corrections allowed for almost 20,000 articles to be transformed economically into the wiki-based online documentation.

An important new requirement was the ability to deliver the documentation at almost any time during the development process. Software documentation is never complete. Therefore it is important to use the existing support of the written content

as early as possible (cp. SCRUM in [Schwaber03]). To ensure high quality throughout the development process, continuous revision was introduced. Every person involved in the development process, including developers and testers, is able to modify the documentation. This drastically improves the documentation quality because more people contribute content. In addition, the low level of effort required to submit modifications increases the likelihood that minor changes such as typo corrections will be carried out.

4 Further Steps

Although ERP systems are assigned to standard software, the installations at customer sites generally differ from each other. Differences can be caused simply by different settings or by customised functions. However, in general the documentation of the installed systems does not vary. The application of the wiki principle enables the customer to tailor the documentation to his/her needs. For instance, a description of the standard input field *material* can be extended by the actual material usage of the customer (e.g. specific grades of steel). These modifications of the customer can be studied by the ERP vendor. Generally, accepted modifications can be transferred to the standard system so customers are integrated into the documentation process. Furthermore, the usage profile of the software can be analysed. Frequently used articles can be an indication for a specific, poorly implemented function. All of these mechanisms help to extend the scope of the documentation process.

References

- [Emery 00] Emery, F.: "Characteristics of socio-technical systems: Introduction to the Concept of Socio-Technical Systems". In: Davis, L. E. und J.C. Taylor (eds.): "In Design of Jobs" Penguin Books, Harmondsworth (2000), 177–198.
- [Gronau et al. 04] Gronau, N., Müller, C., Uslar, M.: "The KMDL Knowledge Management Approach: Integrating Knowledge Conversions and Business Process Modeling" In Karagiannis, D., Reimer, U. (eds.): "Practical Aspects of Knowledge Management" Springer Verlag, Berlin, Heidelberg (2004) 1-10
- [Leuf and Cunningham 01] Cunningham, W.: "The Wiki Way - Quick Collaboration on the Web" Addison-Wesley, Boston (2001).
- [Müller et al. 05] Müller, C., Bahrs, J., Gronau, N.: "Evaluation of KMDL Models of Knowledge intensive Business Processes in the Area of Software Engineering" Proceedings of I-Know '05, Graz (2005), 365-372.
- [Schwaber 03] Schwaber, K.: "Agile Project Management with Scrum" Mircsoft Press, Washington (2003).
- [Sveiby 97] Sveiby, K. E.: "The new organizational wealth: managing & measuring knowledge-based assets" Berrett-Koehler, San Francisco (1997).
- [WikiEngines 05] WikiEngines/ComparingWikis. <http://tavi.sourceforge.net/WikiEngines/ComparingWikis> (2005).
- [WikiMatrix 06] WikiMatrix.org Compare them all. <http://www.wikimatrix.org/> (2006).

[Wohlgemuth 91] Wohlgemuth, A. C.: „Das Beratungskonzept der Organisationsentwicklung“
Haupt, Bern, Stuttgart (1991).