Experiences from Establishing Knowledge Management in a Joint Research Project

Sebastian Meyer[1], Anna Averbakh[1], Torsten Ronneberger[2], and Kurt Schneider[1]

1: Software Engineering Group, Leibniz Universität Hannover

Welfengarten 1, 30167 Hannover, Germany

{sebastian.meyer|anna.averbakh|kurt.schneider}
@inf.uni-hannover.de

2: Audi AG

85045 Ingolstadt

torsten.ronneberger@audi.de

Abstract. Joint research projects are create new knowledge and lessons learned from experience. A research project with several partners is a challenging environment for systematic reuse of knowledge and experience. Knowledge management is often considered overhead, with several tasks added to the workload of the project. This overhead can become overwhelming, since partners from academia and industry have different backgrounds, and may associate different goals and priorities with the project. Industry partners tend to follow strict security guidelines that hamper experience exchange. An extension of project duration is not possible in many publicly funded joint projects. In this paper, we describe our experiences from the initial phase of a major German joint research project with partners from academia and industry. We describe the applied techniques and the lessons learned during the first year the project. We derive conclusions and provide suggestions how to introduce knowledge and experience management in similar projects.

Keywords: experience report, case study, joint research project, knowledge management, experience lifecycle

1 Introduction

Over the last years, research and development (R&D) has become more and more important in Germany. Funding for R&D rose from 20.2 billion EUR in 1982 to 62.2 billon EUR in 2007. Building up researching experience is an important factor in global competition.

Research initiatives usually consist of several smaller research projects. Because of innovative content and the different participants with different knowledge backgrounds, these are very complex projects [1]. This complexity poses a great challenge to project management. As knowledge is one of the most valuable results and assets of a research project, knowledge and experience management is needed to support knowledge generation, transfer and application throughout all organizational and procedural solutions.

An ideal process for collecting and storing knowledge or experiences and making them accessible again is denominated by the experience lifecycle [2] as shown in Figure 1. In this model experiences are collected, structured and stored. After this they are made accessible and can be activated to help in the future. This, of course, leads to new experience, starting a new iteration of the lifecycle.

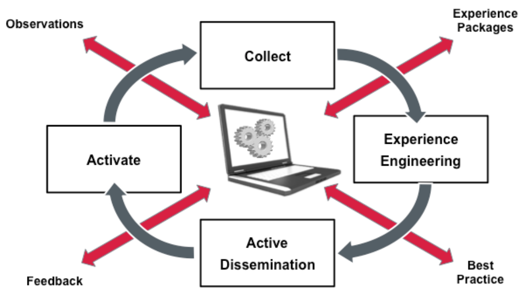

Figure 1: Experience Lifecycle as described in [2]

In this paper, we show an instantiation of the experience lifecycle in a joint research project. We evaluate the experiences made during the first year of the project and derive lessons learned to take into account for a better instantiation.

The paper is structured as follows: Chapter 2 gives an overview of the general characteristics of a joint research project and the challenges for knowledge management. Chapter 3 relates our findings to existing literature. In chapter 4, we present the *e performance* project as a case study for a joint research project and show our instantiation of the experience lifecycle. After this, we give an evaluation of the usefulness and the appropriateness of the chosen result in chapter 5. The evaluation is based on experience made during the first year of the project. Chapter 6 summarizes the lessons learned during the project and presents recommendations for establishing knowledge management in similar joint research projects. We discuss our findings in chapter 7 and give an outlook to further work in this area.

2 Joint Research Projects

2.1 Project Management Characteristics

A joint research project is one frequently used option to close the gap between academia and industry. In the majority of cases, scientists and developers with various specializations and from different organizations are working together towards a defined research goal. They are distributed across different sites. Project members often have different R&D methodologies. They may even stem from different domains as well as disciplines. To effectively work as a project team, knowledge differences must be minimized to assure effective work during the project. Additional time is needed to minimize knowledge differences. That is often neglected or forgotten when partners agree on a project schedule. Project management must, therefore, adapt schedules during the whole project, opening opportunities for team building and installation of infrastructure. It is typically also distributed among different organizations.

Another essential characteristic of a joint research project are the different goals of each project partner. While the overall goal of the project is commonly agreed upon in the project description, each partner can strive after one or more sub-goals, aligned to their overall business. Although project management is in charge of overseeing the agreed goals, partners can distribute their manpower in order to achieve their own high priority sub-goals. Sometimes, reaching a sub-goal conflicts with working towards overall project goals.

2.2 Challenges in Knowledge and Experience Management

Having knowledge in a specific domain is an important competitive advantage. On the one hand, working together successfully in a joint research project depends on knowledge exchange between all contributors. On the other hand, sharing of knowledge with possible competitors or contractors is a fine-grained, often difficult process with many constraints and restrictions. Thus, a fundamental task of knowledge management in joint research projects is to motivate all knowledge carriers to share their knowledge in the context of the project and create the awareness for knowledge as a common resource in the project. Of course, this needs to be supported by the project partner organizations. They need to create a friendly knowledge-sharing environment, even among competitors and contractors working in the same project. There is an obvious tension.

Sharing organizational knowledge among all team members is a popular starting point for creating such an environment. Organizational knowledge can be project plans, competencies and roles of the project members, and contact information. This knowledge should already be in place when the project starts. It is not a competitive advantage, and it can send a signal that sharing knowledge freely is a basic concept of the project.

A much more difficult process is to level the different knowledge backgrounds. To establish a common ground of understanding, it is often necessary for different project members to share their competitive knowledge. As stated above, this is a critical case, since knowledge equals business value for most of the organizations.

At last, it is the function of knowledge management to encourage externalization of tacit knowledge so that the created knowledge will not be lost after the end of the project. Since knowledge management is not the main task for the project members, it is often seen as overhead work for which little time is dedicated.

3 Related Work

Knowledge management in general. Original research on knowledge management is numerous. Polanyi [3] and Nonaka [4] coined the terms tacit and explicit knowledge. Tacit knowledge, like insights or intuitions, dwells in human mind and is hard to convey to others. Explicit knowledge can be expressed and shared in documents. The key to successful knowledge management is finding a way to transform tacit into explicit knowledge. Academia and industry have tried to implement tacit and explicit knowledge transfer. A well-known concept is the Experience Broker introduced by Ericsson [5]. An Experience Broker acts as communicator over projects providing support to other project members. Wenger [6] introduced communities of practice as a social approach to knowledge management and organizational learning. A fundamental work on knowledge and experience management is the Experience Factory by Basili [7]. An experience factory is a separate "logical and/or physical organization that supports project developments by analyzing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand" [7]. These experiences are stored. The knowledge management initiative in the presented case study has been set up according to the Experience Factory concept.

Knowledge management in cooperative projects. Nonaka et al. define a fundamental model (SECI) of a knowledge life cycle in an inter-organizational context, including a combination step, i.e. engineering explicit knowledge into more structured knowledge [8]. This model is a basis to our contribution. Jolly [9] and Soekijad [10] discuss decision and knowledge sharing problems in joint ventures but without giving a pragmatic solution. Trust between (R&D) cooperation partners as an important factor for knowledge sharing has been reported by Inkpen and Curall [11] as well as by Caloghirou et.al. [12]. Concerns of being dependent on the knowledge of joint venture partners are also discussed [13], though giving no implications for project management. Examining research joint ventures, Revilla proposed a taxonomy for knowledge management processes in such a setting [14]. It differentiates types of shared knowledge generated by the research joint venture. It is a conceptual framework and does not consider project management challenges in relation to knowledge management. Bhandar [15] describes how opportunity, motivation and ability can address conflicts like different business interests in collaborative projects and improve knowledge contribution and assimilation. He does not mention technical knowledge management challenges in joint ventures. Kastelli [16] mentions negative influence of rivalry and opportunism on organizational knowledge creation in cooperative R&D projects not giving solutions.

Experience reports. In the past decade, a lot of experience reports on experience and knowledge management initiatives were published [17], [18], [19], [20], [21], [22], [23] in the field of software engineering. They describe company-wide and successful implementations, providing recommendations on how to success in this initiative. Akhavan [24] reports of an unsuccessful knowledge management initiative, mentioning problems like lack of cooperation and knowledge culture, absence of time and an unsuitable infrastructure. This was a company-wide initiative having different reasons for failing and conclusions. It does not give recommendations how to solve these shortcomings. Bass et al. [25] present experiences from a cooperative project on creating a knowledge base. This contribution concentrates on people- and communication-related aspects of distributed software development and does not consider hindrances to the knowledge life cycle.

To distinguish from related work, our experience report presents challenges in conducting a knowledge management initiative in a cooperative research project with industrial and academic partners. We present lessons learned on technical, social and content-related knowledge management problems. From these lessons learned we derive concrete and practical recommendations for similar project settings.

4 Case Study

The following chapter describes our instantiation of the abovementioned Experience Lifecycle on a concrete project. We describe the *e performance* project in the first section of this chapter. Based on the project, we will show how we planned to instantiate the Experience Lifecycle in the first instance.

4.1 Project Description

The *e performance* project is a joint research project that consists of industrial and academic partners with duration of three years. The goals of the project are to research innovative concepts to electrify the power train and to get the implication it has to a complete vehicle. According to this, AUDI AG (Audi), Audi Electronics Venture GmbH (AEV), Rheinisch-Westfälische Technische Hochschule Aachen (RWTH) and Bosch Engineering GmbH (BEG) cooperate to achieve these goals and to establish know-how in the field of electro mobility. These distributed partners have different competencies and specific experience in the development of conventional vehicles and in the area of electro mobility. These various characteristics have been brought together and have been applied to the goals of the project and to get an integrated concept for an electric car.

The *e performance* project is structured so that the characteristic components of an electric car can be analyzed in the work packages (WPs) separately. The WPs themselves are organizational units. They operate in parallel to each other. Agility inside work packages is encouraged to create the innovative concept ideas. In this context, agility denotes the operation of the teams without general process guidelines. This independency gives the WPs room for creativity. Different WPs can work together at the various concepts of vehicle components more easily in a distributed environment. In spite of the agile approach, the developed concepts must pass through predefined basic phases: conception, development/calculation, construction, production, integration/testing. Afterwards, the tested electric car concepts is assembled to a complete vehicle demonstrator to present an integrated solution based on individual solutions.

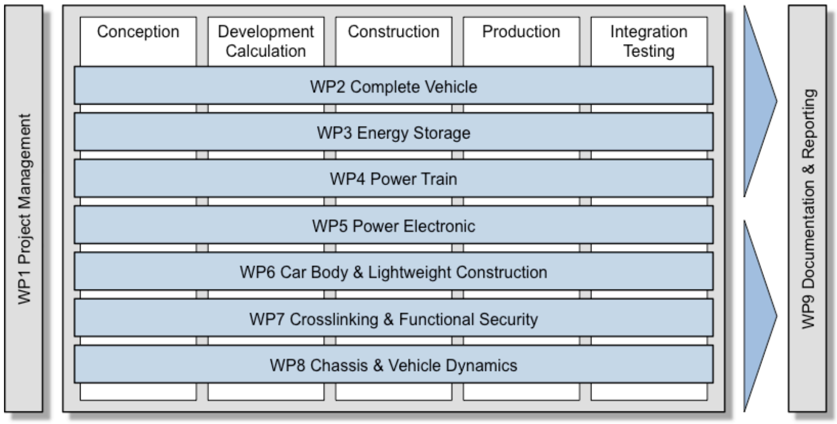

Figure 2: Structure of the e performance project

The project structure is divided into 9 WPs as shown in Figure 2. WP 1 project management and WP 9 documentation and reporting are cross cutting units. The substantial work about the development phases will be made by WP 2-7. The knowledge management is assigned to WP 9. WP 2 complete vehicle has special role. This package coordinates the overall parallel activities of the other WPs and consolidates the integration concepts of the complete vehicle.

4.2 Instantiation of Lifecycle Phases

Figure 3 shows our knowledge management architecture for instantiating the Experience Lifecycle. A Wiki system was established as a central knowledge base. We used an integrated System based on Jira, Confluence and a WebDAV based document storage to implement our instantiation. This system is hosted at a project partners site and already in productive use. We will reference to it as the *productive system* throughout this paper.

For the first iteration, we started at the *collect* step. Since the main objective of the project members is doing research, they cannot afford to devote a lot of additional time to knowledge management. To create just minimal additional workload, we first looked at those documents that are supposed to be created during the project, regardless of knowledge management. These documents were evaluated whether they are suitable for knowledge management. We found the following two document types:

- **The project member list**

  The project member list contains all project participants. It lists each person with his respective contact information and his assignment to the organizational parts of the project. We used this document as a starting point for our FlowMaps [26] and for automatic notifications that are described in detail later on.

- **Meeting protocols**

  To create a protocol for each project meeting is mandatory. A meeting protocol lists all participants, discussed topics and assigned tasks during the meeting.

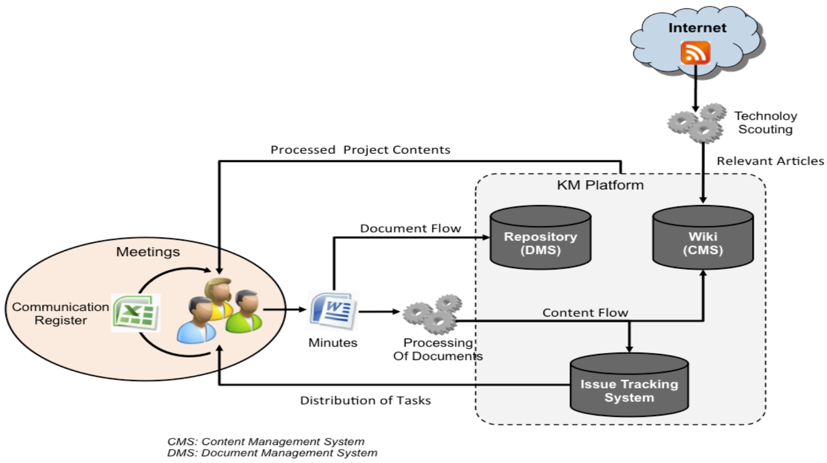

Figure 3: Instantiation of the Experience Lifecycle

We used these two document types in the *collect* step of the experience lifecycle. Since documents of these types have to be created independently from knowledge management, it supports our goal to create just a minimal additional workload for the project members. We created protocol templates to facilitate their automatic processing.

The collected data about project organization, spatial distribution of project members, their contact information and assigned tasks were used for the second step of the experience lifecycle, the *experience engineering*. Here, the collected data has to be prepared and structured, generating new or newly arranged data.

Our infrastructure was designed to create a ticket in an issue tracking system for each assigned task. To add additional information (e.g. contact information) to a ticket that is not part of the protocol, the project member list is used. Additionally to the ticket creation, we create Wiki pages for the information that are contained in the protocol. These pages are grouped by topic and information types (tasks, information, etc.). People that are recognized based on the project member list, are linked to their respective Wiki sites. This is also true for already existing topic-specific sites.

Creating a linked structure from the extracted information is one of the main benefits gaining from using a Wiki. Since the link creation happens automatically, it generates additional value to already existing documents, while creating minimal additional overhead for the people.

During the *active dissemination* step the prepared data is actively distributed to all interested project members. In our instantiation, this phase starts when the Wiki pages and tickets have been created. Both systems – Wiki and issue tracking – inform appropriate members about the creation of new or updated content, to help the dissemination of the generated data. Additional material like presentation slides or CAD data is uploaded by the users to a shared project repository that is subject to access control. While access control may be hindering knowledge distribution, it is necessary in a joint research project to give each project partner sufficient control over their own data. They must be able to decide what they consider sensitive information. Access to relevant documents for the whole project is usually granted to all project members.

The last step in a complete lifecycle is the *activate* step. Since all collected and reorganized data is made available in the used systems (Wiki, issue tracking system, repository), each project member can access the needed data directly. The structuring of the data (linking, categorization) allows filtering and sorting the amount of data adequate for user needs, allowing a quick access to the needed information as well as directed search.

5 Evaluation

After the described instantiation of the Experience Lifecycle, we evaluated user acceptance of the knowledge management initiative. We turned the attention to the Wiki as central information storage, since we noticed that the Wiki's content did not grow as expected. We conducted a survey of 21 persons from all main project partners, most of them leaders of a working package.

Regarding the Wiki, we asked the following questions:

- **Question 1**: Do you use the Wiki?
- **Question 2**: If yes, how often do you use the Wiki?

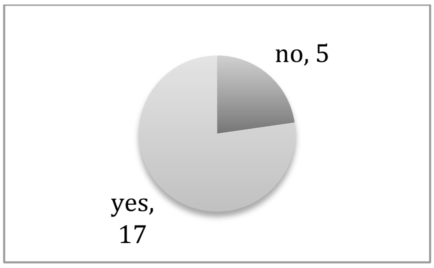

Figure 4: Do you use the Wiki?

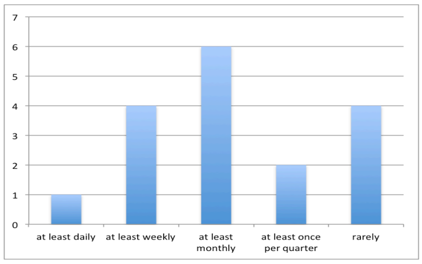

Figure 5: How often do you use the Wiki?

The survey revealed that 17 out of 21 (approx. 81 %) users have at least used the Wiki once, as shown in Figure 4. From these 17 people, only five (approx. 29 %, i.e. approx. 24 % of all asked people) used it at least weekly (Figure 5). Of course, five people using the Wiki regularly is a ratio far too low in a research project.

To better understand why people were not using the Wiki, we asked them for their reasons. We then used Open Coding [27] to classify the answers. In the end, we derived three main categories of reasons from analyzing 65 answers: *Technical issues*, *social issues* and *issues directly related to knowledge management*. We will discuss each category and its sub-categories in the following sections.

5.1 Technical issues

This category contains all issues caused by technical infrastructure. One of our main goals, when planning the instantiation, was to establish knowledge management without adding large overhead, i.e. minimizing the additional workload for the project members. To achieve this goal, the tools were supposed to support users to document their knowledge and use already documented knowledge. This category consists of two sub-categories:

- **System too slow**

  Project participants rated the system as generally too slow. This was especially the case for the shared document repository, when working with numerous and large files (e.g. CAD data).

  The logon to the system was also rated too cumbersome. This is true for the logon performance mechanism and for the time needed to initialize the additional security mechanisms like one-time password tokens.

  Conclusion: Users that have to work with a system they consider too slow, will lose their interest in the system. They will start to avoid the system for smaller tasks and only use it, when it is absolutely necessary.

- **No access to system**

  Another technical problem is access restriction to the system. Lacking an (activated) account, users complained about not being able to access the system. Since we implemented the infrastructure on a project partners' productive system, clearance for accounts have to be done through their clearance process.

  Using a productive system makes the process of account clearance for other organizations a security relevant process. It has to be guaranteed that other internal projects cannot be accessed. Even if there are organizational reasons for a slow response to account applications, users only notice the missing access. To bypass the time without a working account, they use other applications.

  Conclusion: If organizational overhead for creating and clearing new accounts is too big, users have to wait too long. They start using another system during this time and lose interest in the original system. By the time their account is cleared, they will be accustomed to the alternative. This may leads to an inconsistent knowledge base, a tedious process to transfer the data, or the avoidance of the original system.

5.2 Social issues

Social issues are triggered directly or indirectly by project members. These issues often hinder people from completing their tasks of knowledge sharing. In our case, we experienced the following four sub-categories of social issues:

- **System is external**

  Establishing the knowledge management environment, we decided to use a project partners' productive environment to host the infrastructure. From other partner organizations' perspectives the system is external. External systems are considered less trustworthy compared to own systems.

  An additional problem is the uncertainty of what happens to the data after the project is finished and who will have access to which data in what format. Since we used a proprietary system, it was not possible for external users to create a full system backup on their own.

  Conclusion: If one project partner hosts the system, at the beginning it is necessary to clearly commit to others what happens to the data at the end of the project. There should also be a possibility for external users to get a full backup of all data at any time.

- **System is too complex**

  Some users found that a system containing many sub-systems (like a Wiki, issue tracking, etc.) is too complex, especially as these several sub-systems were not integrated. This leads to ambiguity and confusion where to put which type of data and where to search for them.

  Conclusion: If several non-integrated systems are used, there has to be a clear guideline about the use of each system for a certain type of data.

- **Not enough time**

  Users having not enough time are a common issue. Since knowledge management is not their main task, they are often not able to spent additional time for it.

  Conclusion: Tools and methods for knowledge management have to be designed to avoid any additional overhead, especially those operated by users whose main task is not knowledge management.

- **Does not fit into workflow**

  Even if the technical infrastructure is in place and all other issues (see above) are fixed, one limitation for productive usage stays: the inadequate workflow. In our case, all project members used e-mails to communicate with each other and to get informed about new content. Contrary to this, the Wiki implements a polling pattern, requiring from the users to ask proactively for new content. This interrupts the users' workflow, restraining instead of supporting them.

An automatic e-mail notification about new content was not helpful. Many users were notified about content that was not relevant to them. The outcome was information overflow letting the users to ignore e-mails from this source.

Conclusion: When installing new systems to support knowledge management, e-mails as a main component of a personal workflow should not be underestimated. E-mail should not be misused to send irrelevant or trivial information to the users.

5.3 Issues directly related to knowledge management

This category of issues is directly related to knowledge management. Contrary to social and technical issues, reasons for KM-related issues can usually be influenced. We found two KM related issues in our project:

- **No useful content** and **no need to use**

  These two issues depend on each other. As long as there is no enforcement to use a certain system, just a few people will use it voluntarily. As only a few people use a system and just a few of them add new information to the system, there will be no useful content for most users. As a consequence, they will still not use it. Another consequence from the lack of participation is old news remaining on the homepage and not being updated.

  Conclusion: A new system needs significant contribution from knowledge management to make it interesting for other users. Otherwise they won't use it.

5.4 Lessons Learned

In this section, we present lessons learned from our instantiation of the Experience Lifecycle and feedback we received from participants. For the next iteration of our instantiation architecture, these lessons have to be taken into account:

- **Fixed productive system**

  Using a productive system, hosted at the site of a project partner was not an ideal decision. Needed adaptions of the system like a special view for agendas could not be implemented, since changing the base system could only be done after passing internal security tests. These tests usually last longer than a month. This is far too long for users to wait.

  High security barriers around a productive system are another obstacle. They effectively prevent automation of tasks, which renders the implementation of additional tools impossible. Those tools could support low-effort knowledge work.

Conclusion: The use of productive systems for systems with unfinished or fast changing requirements is not adequate.

- **Volatile content**

  Since the described project is a joint research project, the documented knowledge can be very short-lived. It may be outdated at a fast pace or should be enriched with new findings. This leads to problems, when trying to synchronize two different sources like documents and the Wiki.

  This problem may be diminished by an automatic synchronization mechanism. However, depending on the complexity of the synchronization and the temporal distance between synchronizations, there may be a difference between actual data in the reference documents and the data in the knowledge base. Having a knowledge base that is filled with outdated or wrong data will lead the user to lose trust in it.

  Conclusion: If reference documents for the data in the knowledge base change regularly, all changes must be adopted to the knowledge base immediately.

- **Unfamiliar Workflows**

  Even though we carefully inspected the workflows and habits of the project members and optimized our tools to work with MS Office, we underestimated the importance of e-mail for the users. The change from the e-mail application into another, usually uncommon application, led to disruption of the workflow and has been therefore done reluctantly.

  Additionally to this, many users experienced knowledge management as an additional task, since it has nothing to do with their regular workflow items. This is true for their main project task (i.e. researching) as well as for using additional tools for knowledge management. Even if the tools look and feel similar to the commonly used tools, they are still subtly different and may yield unexpected results.

  Conclusion: Inspecting workflows, it must be considered that commonly used tools cannot easily be replaced by similar tools that may be subtly different. E-mail must be considered as the most important utility.

6 Recommendations for a better start

Based on the abovementioned lessons learned, we adjusted our instantiation of the Experience Lifecycle to antagonize the discovered issues. This chapter describes the adjustments and gives recommendations for similar projects.

- **A tool to collect data**

The choice to use meeting protocols as a data source for the knowledge base was good, but not sufficient. We wanted project members to describe additional experiences and problems made during their daily work.

We found that problems will only be documented at the time they occur. They will not be documented afterwards. To support documentation, we created an iPad application which is connected to our issue tracking system. As an advantage of this solution, project members can carry the tablet with them and document problems and experiences as they go, optionally adding additional media like photo. Since the application runs locally on the Apple iPad, this can also be done in cases of no Wi-Fi. Lack of Wi-Fi is common in secure manufacturing areas.

Recommendation: If the documentation of problems is desired, project members need a tool to enter the problems directly as they happen. This tool must be able to completely capture the documentation, making it possible (but not necessary) to rework the documentation.

- **Using a version control system as project repository**

  After getting feedback about the shared project repository in use, we decided to introduce the Apache Subversion version control system instead. Apart from the automatic versioning of Subversion (which was not implemented in the old repository), another benefit is the possibility to get a complete copy of the contained documents at any time. The first feedback we received points to a much higher acceptance of the Apache Subversion than of the old system.

  Compared to a Wiki, the ability to create linking structures between the documents is lost. As compensation, we plan to implement a novel web platform, which will show categorized links to the documents. This should also eliminate the problem of outdated content, since the content is always contained in the reference documents and must not be transformed.

  Recommendation: If the content of a knowledge base is transformed from volatile reference documents, it is better practice to directly link to the documents and structure the links instead.

- **Better workflow integration**

  After getting feedback about the integration of our methods into the everyday workflow of project members, we stopped working with similar tools and concentrated on supporting the tool already in place.

  Recommendation: When introducing new tools into existing workflows, it has to be precisely checked if already existing tools can substitute the new tools. In this case not using the ideal tool is often better. The familiarization process is then much shorter leading to higher productivity and more satisfied project members.

7 Discussion and Outlook

In this paper we described our experience made in the first year of establishing knowledge management in a joint research project. In the beginning we instantiated the Experience Lifecycle. The issues related to this instantiation were then inspected through interviews with the project members.

We described experiences and lessons learned during this year in detail. We hope to give similar project settings a better starting point to plan an instantiation of the Experience Lifecycle. These are our recommendations in short:

- Do not use a productive system as a knowledge base.
- Synchronize changes in referenced documents immediately to the knowledge base.
  - If this is not possible, just link to the original documents.
- Take already established workflows into account and try to support existing tools instead of introducing similar ones.
- Provide the possibility to document problems immediately when they occur.

From this analysis, we received a better awareness about the necessary adaption of our Lifecycle instantiation.

To prove the usefulness of our adaptation to the needs of the users, we will have a look at the usage pattern for this new approach. We expect the users to work more often with the new infrastructure, since it better supports their native workflow. Preliminary results from surveys concerning the changes indicate a much better acceptance rate.

Another area of research is structuring the data in the version control systems. For this, we plan to use the linking categorization to the documents and extract an ontology from them. This will help us to automatically categorize new documents as they are uploaded to the system. It also allows us to integrate a semantic driven search functionality for documents. Another use of the categorization is to generate comprehensive documentation from already existing documents.

8 References

1. Arranz, N., Fdez de Arroyabe, J.C.: Complex joint R&D projects: From empirical evidence to managerial implications. Complexity. 15, 61-70 (2009).
2. Schneider, K.: Experience and Knowledge Management in Software Engineering. Springer Verlag (2009).
3. Polanyi, M., Sen, A.: The tacit dimension. Peter Smith Gloucester, MA, Garden City, NY (1983).
4. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press (1995).
5. Johannson, C., Hall, P., Coquard, M.: Talk to Paula and Peter - They are Experienced. International Conference on Software Engineering and Knowledge Engineering (SEKE 99). Workshop on Learning Software Organizations (1999).

6. Wenger, E.: Communities of Practice - Learning, Meaning, and Identity. Cambridge University Press (1998).
7. Basili, V.R., Caldiera, G., Rombach, H.D.: Experience factory. Encyclopedia of software engineering. 469-476 (1994).
8. Nonaka, I., Toyama, R., Konno, N.: SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation. Long Range Planning. 33, 5-34 (2000).
9. Jolly, D.: Sharing knowledge and decision power in Sino-foreign joint ventures. Asia Pacific Business Review. 9, 81–100 (2002).
10. Soekijad, M.: The competitive factor in knowledge sharing networks. Presented at the (2002).
11. Inkpen, A.C., Currall, S.C.: The coevolution of trust, control, and learning in joint ventures. Organization Science. 586–599 (2004).
12. Caloghirou, Y., Vonortas, N.S., Ioannides, S.: European collaboration in research and development: business strategy and public policy. Edward Elgar Publishing (2004).
13. Walker, D.H.T., Johannes, D.S.: Preparing for organisational learning by HK infrastructure project joint ventures organisations. Learning Organization, The. 10, 106–117 (2003).
14. Revilla, E., Sarkis, J., Acosta, J.: Towards a knowledge management and learning taxonomy for research joint ventures. Technovation. 25, 1307-1316 (2005).
15. Bhandar, M.: A Framework for Knowledge Integration and Social Capital in Collaborative Projects. Electronic Journal of Knowledge Management. 8, 267–280.
16. Kastelli, I.: Organisational Knowledge Creation in the Context of R & D cooperation . The role of absorptive capacity. Knowledge Creation Diffusion Utilization. (2006).
17. Thomas H. Davenport David W. De Long, M.C.B.: Successful Knowledge Management Projects. Sloan Management Review. (1998).
18. Jørgensen, M., Sjøberg, D.I.K., Conradi, R.: Reuse of software development experience at Telenor Telecom Software. In: - (ed.) European Software Process Improvement Conference (EuroSPI'98). pp. 10.19-10.31. -, Gothenburg, Sweden (1998).
19. Dingsoyr, T., Conradi, R.: A survey of case studies of the use of knowledge management in software engineering. International Journal of Software Engineering and Knowledge Engineering. 12, 391–414 (2002).
20. Brössler, P.: Knowledge management at a software house: An experience report. Learning Software Organizations. 1756, 163–170 (2000).
21. Feldmann, R.L., Pizka, M.: An On-line SE Repository for Germany's SME - an experience report. Advances in Learning Software Organizations. 4th International Workshop (LSO 2002) (2002).
22. Houdek, F., Bunse, C.: Transferring and evolving experience: A practical approach and its application on software inspections. Learning Software Organizations. 1756, 210–226 (2000).
23. Schneider, K., Schwinn, T.: Maturing experience base concepts at DaimlerChrysler. Software Process: Improvement and Practice. 6, 85–96 (2001).
24. Akhavan, P., Jafari, M., Fathian, M.: Exploring failure-factors of implementing knowledge management systems in organizations. Journal of Knowledge Management Practice. 6, 1–9 (2005).
25. Bass, M., Herbsleb, J.D., Lescher, C.: Collaboration in global software projects at siemens: An experience report. Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on. pp. 33–39. IEEE (2007).
26. Stapel, K., Knauss, E., Schneider, K., Zazworka, N.: FLOW Mapping: Planning and Managing Communication in Distributed Teams. Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on. pp. 190–199. IEEE (2011).
27. Corbin, J., Strauss, A.: Open Coding. In: Corbin, J. and Strauss, A. (eds.) Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. pp. 101-121. Sage Publications, Thousand Oaks, CA (1998).