

# Tool Support for Inter-Team Learning in Agile Software Organizations

Thomas Chau, Frank Maurer

University of Calgary  
Department of Computer Science  
Calgary, Alberta, Canada T2N 1N4  
{chauth,maurer}@cpsc.ucalgary.ca

**Abstract.** The need for organizational learning support is common among all software development companies but is not addressed by agile software methods practitioners. The typical Experience Factory approach to address organizational learning in software companies often stresses structured and explicit knowledge as well as controlled learning mechanisms. In contrast, the Communities of Practice approach relies more on tacit knowledge and self-organization. To bridge the gap between these two conflicting approaches, this paper proposes a suite of lightweight knowledge sharing tools by which software organizations are given the flexibility to control how structured and controlled their learning processes are. Initial results from early evaluations of the proposed tools are also given.

## 1 Introduction

Software development is a knowledge-intensive process requiring expertise in the customers' business domain and mastery over technological skills. These knowledge requirements are common among all software development organizations but are even more demanding in those "sweet spot" business environments for agile software methods practitioners. This is due to the fact agile teams are often confronted with short development and learning times as well as unpredictable and continuous changes in both technologies and customer needs. Furthermore, for any software organizations to survive in the long term, it is imperative that they apply experience gained from past projects to similar projects in the future. This underlies the need for knowledge sharing and organizational learning support in software organizations, and among agile practitioners in particular. We use the term "Inter-team Learning" to indicate knowledge sharing across team boundaries without moving people between teams.

One approach developed by the software engineering community to address the knowledge needs and overcome the learning challenges mentioned above is the Experience Factory concept. One of its intentions is to package "experiences by building informal and formal product models...and other forms of knowledge via people, documents, and automated support" [1]. While the Experience Factory approach does not preclude the use of tacit knowledge and informal learning processes, many of its

existing implementations rely more on formal and explicit knowledge for knowledge transfer [19] and controlled learning mechanisms [18].

Encoding knowledge in structured and explicit form has the benefits of mitigating risks of knowledge loss and enabling distributed teams to collaborate in a time- and space-independent manner. It also makes knowledge available in a more reusable format if used together with controlled learning mechanisms such as formal training and organizational units that conceive and distribute knowledge to other operational units. The overall effect can help reduce duplicated effort in devising similar solutions and prevent past failures from recurring. However, these benefits are best realized in a stable business environment where there is limited variety to development tasks – not a typical scenario for agile teams. In addition, the heavy emphasis on structured and explicit knowledge over tacit knowledge and controlled learning processes over self-organizing processes contradicts agile principles [2]. An experience factory is an organizational unit that gathers experiences, makes them accessible to others (e.g. in the form of “best practices”) and provides line units with support in adopting these best practices. A major risk of such a centralized approach is that its team members are often not involved in everyday software development and might lose contact with reality over time. As a result, Experience Factories or Software Process Groups are sometimes perceived by the “real” software developers as living in an ivory tower.

Given the incompatibilities between the Experience Factory approach and agile principles, how can software organizations facilitate knowledge sharing and organizational learning while remaining agile in the process? In an attempt to overcome these incompatibilities, this paper proposes a suite of lightweight knowledge management tools, MASE and EB, by which software organizations are given the flexibility to choose how structured and controlled their learning processes are.

Another approach originated in the field of organizational dynamics to address issues in organizational learning and knowledge sharing is called Communities of Practice [6]. Unlike Experience Factories, Communities of Practice are informal and self-organizing networks of professionals. They are intended for (1) helping capture and share tacit knowledge that formal system cannot capture as easily [5]; (2) reducing time seeking for expertise; (3) linking staff interested in similar knowledge domains but dispersed across organizational units; (4) connecting and coordinating isolated initiatives addressing similar knowledge areas [7]; (5) and retain talents [3,4]. Furthermore, their people-centric approach is in line with the agile principles. Despite these benefits, potential drawbacks to the Communities of Practice approach include mismatch between the communities’ knowledge domains and the business’ core competences as well as stagnation due to overly tight social relationships among community members that may stifle new ideas and constructive criticisms [7].

## **2 Tool Criteria**

Instead of viewing the two conflicting approaches as a dichotomy, we concur with others [8,10] that learning processes, in practice, often lie somewhere on a continuum from highly structured processes like the Experience Factory approach at one extreme to highly unstructured processes as in the Communities of Practice approach at the

other extreme. One implication of this perspective is that knowledge sharing tools need to support: (1) formal learning based on prescribed processes and (2) informal situated learning that are often found in day-to-day collaborative work. The fact that people move unconsciously “continually and effortlessly among different collaboration styles: across time, across place, and so on” [9] also demands knowledge sharing tools to accommodate more than one collaboration style like: (3) co-located and distributed teams; (4) synchronous and asynchronous activities; and (5) use of structured and unstructured information. For such tools to be useful for agile organizations they also need to support (6) the social context critical to nurturing a knowledge sharing environment - providing information needs to be as easy as accessing it and expertise can be exchanged in informal and spontaneous manners.

### **3 The Tools: MASE & EB**

These requirements are met by our tools, MASE and EB. MASE is a process support system to be used by agile teams for project planning coordination purposes whereas EB is an experience base to be shared among all teams in the organization.

#### **3.1 Support for Co-located and Distributed Teamwork**

Both EB and MASE are implemented using Web technology which makes the tools accessible anytime anywhere by users with a browser in their computing environment. The tools do not distinguish users working at the same place from those who work at different places. Hence, both EB and MASE are capable of supporting collaborative work for both co-located and distributed teams.

#### **3.2 Support for Unstructured and Structured Information Content**

Both EB and MASE allow information to be externalized into highly unstructured as well as structured formats. Further, they make writing – providing information – nearly as easy as reading – accessing information. Unstructured information usually consists of text and graphics whereas structured information is typically stored in a database and must follow a schema. To support unstructured information content, both EB and MASE use Wiki technology for their user interfaces [12]. Wiki technology enables any users to access, create, structure, and update web pages in real-time using only a web browser. These web pages, also known as wiki pages, act like electronic bulletin board discussion topics, each with a unique name. Users apply the Wiki markup language to edit wiki pages. Wiki markup is much simpler than HTML: a list of all Wiki markup command including examples fits onto a single page.

One may argue that wiki pages are no different from any other document and will suffer from the same maintenance problems. Wiki technology mitigates this risk by automatically creating links from a wiki page to particular topic pages if the names of those topics are mentioned in that page. This helps minimize the users’ effort in maintaining the relationship among the content in different wiki pages and enhances

knowledge discovery. These benefits, however, are best realized if users adhere to the same terminology when contributing content to wiki pages.

Information content in a wiki page is all free-formatted text. This is not the case when users try to update a typical web page. Normally, the information content on a web page that users see is embedded among presentation information such as HTML markup elements. For the users to edit such a web page, they need to spend the extra effort to first extract the information content then begin the actual editing of the content. Sometimes, this additional effort is so time-consuming that the users give up on editing the content, thus causing knowledge content to degenerate over time. The fact that information content of web pages in EB and MASE is in free-formatted text facilitates efficient collaboration between knowledge contributors and readers.

To support structured information content, EB and MASE employ a library of plug-ins that store specific data in a database. A plug-in in EB or MASE is either presented as an input form that allows users to submit information or as a table that displays information retrieved from the database in a structured fashion. Users can include such plug-ins in any wiki pages simply by referencing their names. Currently, the libraries of plug-ins in EB and MASE include those that are specific to agile teams and generic collaboration tools like rating a specific wiki page which supports collaborative filtering. The fact that any content on any wiki pages is modifiable and that any plug-ins can be included in any wiki pages gives users the flexibility to control how structured or unstructured they want their organizational memory to be.

Storing information does not guarantee that others can find it. Retrieving information from a repository that combines structured and unstructured data often requires users to learn two different query mechanisms. To overcome the resulting usability problems, MASE provides a single full-text searching capability on any unstructured and structured content.

### **3.3 Support for Personal Portals**

When a team member first logs into EB or MASE, she is automatically provided a personal wiki page that serves as her personal portal. She can store in her portal any content, either structured or unstructured information (or both). The content may be relevant only to her or to some subset of team members; it may not even be related to the project or task at hand. The key idea is that a team member has complete control over the type and granularity of the information content that she deems is relevant for her. Like any wiki pages, the content in the personal portal is accessible by others.

### **3.4 Support for Asynchronous and Synchronous Collaboration**

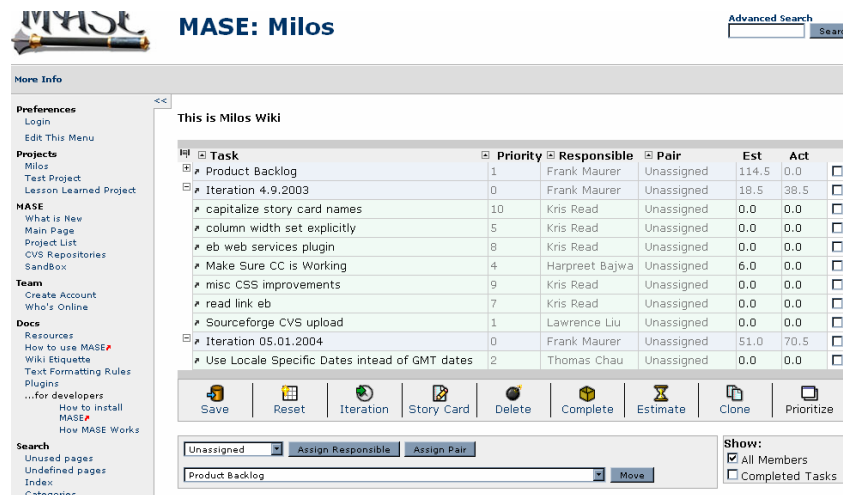
Both EB and MASE support asynchronous collaboration by storing the state of any wiki pages one has worked on before logging out. This resembles the common practice in the real world where team members leave artifacts in a physical place for others to review or update when they work at different times.

EB and MASE also support synchronous work through integration with a real-time collaboration tool, Microsoft NetMeeting. Every time when a team member logs into

EB or MASE, the tools track the network address of that team member's computer. This allows EB and MASE to display which members of a team are currently using the system, thus making all online team members aware of each other's presence. This allows for team members to easily establish informal, synchronous and spontaneous communications with one another – most likely moments when sharing of tacit knowledge take place [13].

### 3.5 Support for Agile Practices

As a process support system for agile project teams, MASE supports agile practices through its library of plug-ins. For project planning, project managers and customers can create iterations and user stories as structured data objects to be kept in the database and accessed through plug-ins. The developers can submit to MASE their time estimates for tasks that they assume. MASE will then suggest to both the development team and customers the appropriate size for the next iteration based on the developers' estimation accuracy from the previous iteration. Using the suggested iteration size, customers can prioritize user stories and move them from iteration to iteration or move them back to the product backlog. During the course of the project, both the customers and development team can track work progress at various granularities using the Whiteboard plug-in (project, iteration, user story) and view effort metrics for a particular individual or for the entire team.



**Fig. 1.** The above plug-in is named “Whiteboard” to mirror the big physical whiteboard often used by agile teams to show vital information that the entire team should know

Through the Whiteboard, developers can see the features and tasks allocated for each of the iterations in the project and track their time. Details of a user story or a task are stored in a wiki page allowing developers to annotate in free-formatted text. Leverag-

ing MASE's integration with NetMeeting, developers can also perform distributed pair programming by sharing their code editor and thus collaborate on a design together using the shared whiteboard. Using the video and audio conferencing and multi-user text-chat features of NetMeeting, distributed team members who work at the same time can perform daily Scrum meetings.

### 3.6 Support for Organizational Learning

To facilitate learning based on prescribed processes, EB allows users to construct models of common tasks. Such process models can contain detailed description about common tasks and their possible decompositions in structured format. Users can also store any specific information about the model that one should pay attention to as free-formatted text. The left window in figure 2 shows the process model for a common development task, EJB Creation and Deployment. It contains descriptions of the steps one should follow in performing tasks of similar kind and some common bad practices to avoid. The process model basically acts as a type system for project tasks.

Depending on how formal and controlled a company wants its organizational learning process to be, the job of process modeling can be open to anyone in the organization or it can be performed by designated roles, such as a group of knowledge brokers or, in the case of an agile organization, a group of Scrum Masters [11]. EB can support either approach.

While it is important to capture knowledge that is useful for the whole organization in process models, this measure alone is insufficient for facilitating learning based on prescribed process. Studies reveal that people often are unaware of information that may be of help to them even if such information exists in the organization's knowledge base [20]. It is therefore crucial to make the organizational knowledge easy to find. To this end, EB delivers the task-relevant information in process models to users' project workspace on demand; thereby reduces users' time in looking up information. This is demonstrated in the right window in figure 2.

It shows that Thomas is responsible for the "Refactor Version EJB" task in the "Milos" project and this task is associated with the "EJB Creation and Deployment" process model in EB. This association between a project task in MASE and a process model in EB can be made by Thomas or his manager. The key idea is that every time Thomas accesses information about the "Refactor Version EJB" task in MASE, he automatically receives information about others' experiences on tasks of similar kind, in this case, EJB Creation and Deployment.

These experiences may have been provided by others, for example Jack and Jill, whom Thomas may know but are from other teams in the organization. Since these experiences are annotated on wiki pages, the open-edit nature of wiki and EB's support for synchronous work allow Thomas, Jack, and Jill who share common interest to establish contact, exchange their individual experience and expertise, and collaborate together even though they may be dispersed across organizational units.

Anecdotal evidence and thriving inter-organizational practice that use Wiki servers suggest that this kind of informal knowledge sharing actually happens [16]. Through its integration with multiple MASE project repositories, an organizational experience base like EB facilitates the establishment of communities of practice.



Fig. 2. The integration between EB (left) and MASE (right) can help disseminate knowledge created collaboratively by members in the organization to different project teams and can link members of different teams who are interested in similar task-related knowledge areas

#### 4 An Explorative Case Study

To evaluate whether a Wiki-based knowledge sharing tool like EB facilitates inter-team learning, an explorative case study of EB is currently being conducted: (1) to determine whether inter-team learning occurs when members of different teams are facing common problems or working on similar tasks; (2) to elicit the challenges and enabling factors for inter-team learning using a Wiki-based knowledge sharing tool;

(3) to identify usage patterns of a Wiki-based knowledge sharing tool, and (4) to gather feedback on the tool's usability and impact on software development.

#### 4.1 Experimental Context

Participants of this study include a total of 42 senior undergraduate students of which 25 are enrolled in the course, Web-Based System, at the University of Calgary (UC) and the remaining 17 enrolled in a similar course, Internet Software Techniques, at the Southern Alberta Institute of Technology (SAIT). The 25 UC students are divided into 6 teams each consists of 4 to 5 members. Likewise, the 17 SAIT students are divided into 6 teams each consists of 3 to 4 members.

Both courses last 4 months and they expose students to the latest technologies and practices in building Web-based enterprise systems. The content of both courses are offered by the same instructor in approximately the same time period. Over the 4-month time frame, all teams are required to complete 6 comprehensive programming assignments involving the construction of a document review system. Each assignment is to be completed in a short iteration, usually 2 to 3 weeks long. Teams are encouraged to share their individual learning and work on their assignments following agile development practices such as short iteration and pair programming.

In the first week of the study, all teams received training on using EB and MASE. When each assignment is released, dedicated KM brokers (the class instructor and teaching assistants) provide in EB information that they deem as important for all teams to know for completing their assignments. All teams have read and write access to EB and they are each given their own project information repository, MASE.

To collect data to meet the goals of the study, EB logs all use of the tool by any participants. In addition, participants will be asked to voluntarily fill out questionnaires at the end of the 4-month observation. The context and dependent variables of this study are summarized in figure 3.

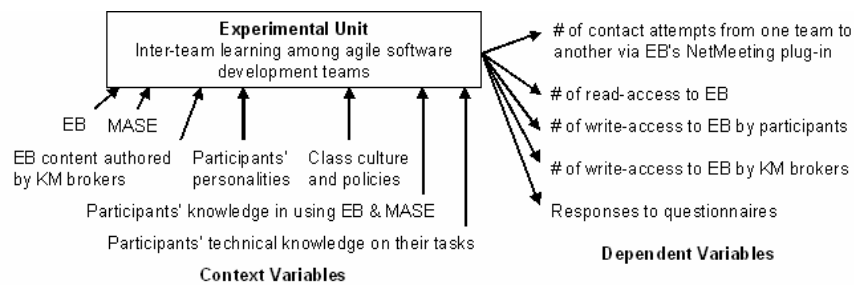


Fig. 3. Experimental Unit

#### 4.2 Initial Results

At the time of this writing, all teams have finished two out of six assignments. About 9 weeks of data on the usage of EB have been collected and over 25,000 transactions



have been recorded. Results shown in table 1 indicates that students use EB primarily to retrieve information they need for their development tasks; as bookmarks to other external online resources; and to share information.

Table 2 indicates that students are contributing more content to EB than dedicated KM brokers (the course instructor and teaching assistants). However, most of the content contributed by students is in the students' personal portals.

**Table 1.** EB Usage Breakdown

<i>EB Usage Breakdown</i>	<i>usage (%)</i>	<i># of transactions</i>
Read	82.38	22040
Bookmark to External Sites	13.16	3521
Write	4.02	1075
Search Query	0.37	98
Collaborative Work using NetMeeting	0.07	20
<b>Total</b>		26754

**Table 2.** Breakdown of content contribution in EB by users' role

	<i>KM Broker</i>	<i>Student</i>	<i>Total</i>
Personal Portal Content	64	406	470
Common Information (e.g. FAQs)	464	141	605
<b>Total</b>	528	547	1075

### 4.3 Interpretation

By inspecting the individual postings made by the students to the shared content in EB, we did observe different teams offering advice to each other's problems. However, since the benefits of inter-team learning are subjective to individual perceptions, we are unable to assert whether a Wiki-based knowledge sharing tool like EB meets its goal of facilitating inter-team learning using only the available initial results.

The high proportion of content contribution on personal portals appears to support the need for providing personal information space in a shared experience base. This assertion, however, requires confirmation from the participants' responses to the surveys which will be distributed at the end of the study.

The high proportion of content contribution on shared content by the KM brokers echoes anecdotal evidence of the need for dedicated KM brokers or a core group of community members to maintain an experience base despite the open-edit nature of Wiki [7].

### 4.4 Limitations of the Study

A number of limitations in this study should be recognized. First, the number of participants in the study is small and the observation period is short. This is going to impact on how representative our findings will be. Second, the lack of objective measures of knowledge exchange will be an inherent limitation of the study.

## 5 Related Work

Existing tools which support agile practices or inter-team learning include VersionOne [14], Xplanner [15], TWiki [16], and BORE [17]. All of them are web-based tools which allow them to be used by team members working at the same place or at different locations. However, they differ in terms of their level of support for the various agile practices, capabilities in accommodating the different collaboration styles, and facilitation for organizational learning.

Both VersionOne and Xplanner support project planning and tracking. VersionOne, in particular, provides each team member with a private web page which serves as his/her own information portal. However, VersionOne pre-defines all the content in one's personal information portal showing only tasks that are assigned to the team member. In fact, all information content in both tools can only be created and browsed in a structured way. Team members cannot control the formality of the content nor can they specify their own search query for retrieving information.

TWiki also supports those agile team-related features provided by VersionOne and Xplanner but its usage is not targeted to agile development teams. It differentiates itself as a collaboration platform, not just a tool. This can be seen in the multitude of team-oriented tools it provides, such as event calendar, action tracker, drawing editor, and vote collection. As the name suggests, TWiki is developed based on the Wiki technology. Hence, TWiki and EB share a lot in common: plug-in architecture, support for unstructured and structured information content, personal portal support, and full-text search. TWiki allows a set of web pages to be grouped together, known as a TWiki Web. This indirectly facilitates the establishment of communities of practice in that a community can have its own TWiki Web. One drawback of TWiki is that it provides no direct support for online team members to be aware of each other's presence. This limits the opportunities for team members to establish informal and spontaneous encounters with one another. TWiki also does not have explicit support for modeling common tasks, making learning based on prescribed processes difficult.

Unlike the other three tools, BORE does not directly support specific agile practices. As an implementation of the Experience Factory concept, it provides a repository that contains experience collected from projects across the entire organization. These experiences are organized as cases, which are used to generate pre-defined tasks for similar projects in the future. A case is similar to a project task in nature. The generated set of tasks serves as a "best practice" guide. Project team members can diverge from the generated plan and not perform the suggested tasks if they deem the tasks to be inappropriate for the project situation at that time. In such cases, team members can submit their experiences and details of the tailored tasks in a structured format to the repository. A dedicated team of people is recommended to maintain the integrity of the cases stored in the repository. BORE's explicit support for the reuse and sharing of process knowledge facilitates learning based on prescribed process. However, it does not possess the support provided by EB in facilitating the informal and spontaneous exchange of experience and expertise in day-to-day collaboration that is critical for situated learning.

## 6 Open Issues and Future Work

Immediate future work includes assessing the impact of the tools on software development and inter-team learning by analyzing the usage data and responses to the perception study once the experiment completes. The next step is to involve companies to engage in further pilot studies to evaluate our approach in realistic settings.

For cultivating communities, the integration between the process models in EB and specific tasks in MASE currently only fosters community of users who are interested in task-related knowledge topics. To identify potential users who may be interested in similar knowledge topics that are not task-related, future work include developing plug-ins that group users by wiki pages that they frequently browse or contribute.

Our approach is designed specifically for agile teams and their situated environments, and so may not be appropriate for other types of projects or environments.

Since organizational learning and knowledge sharing is not just a technical but also a social problem, a major open research question is the organizational changes needed to foster the knowledge sharing culture that is essential for our tools and any other knowledge management tools to realize their benefits.

## 7 Conclusion

The knowledge-intensive nature of software developments underlies the need for knowledge sharing and organizational learning support for all software companies. Agile software organizations are no exception, especially if they are to survive in the long term. Typical Experience Factory approach to address organizational learning often stresses on the use of structured and explicit knowledge as well as controlled learning mechanisms. While this approach supports learning based on prescribed processes well, it contradicts agile principles. In contrast, the Communities of Practice approach's valuation on tacit knowledge and self-organizations is a good fit with agile software methods but this approach also has some inherent drawbacks. This paper describes a suite of lightweight knowledge management tools, MASE and EB, which leverages on the strengths of these two conflicting approaches and bridges their incompatibilities.

On one hand, EB supports the Experience Factory approach by allowing one to construct process models containing structured knowledge on common tasks. Through its integration with MASE, a process support system for agile teams, EB can deliver task-relevant information to knowledge workers' project workspace on demand, thereby facilitating learning based on prescribed process. On the other hand, EB also facilitates informal situated learning often found in communities of practice through its integration with MASE and its ability to accommodate different collaboration styles: co-located and distributed teamwork; asynchronous and synchronous work activities; and use of structured and unstructured information.

While the approaches of Experience Factory, Communities of Practice, and Wiki-based knowledge management tools are not new, the novelty of our approach is the integration of these concepts and tools to provide software companies a flexible mean to support their organizational learning and knowledge sharing processes, however

controlled or structured they may be. By integrating these various approaches, we hope to get the most benefits without compromising the agility of software firms.

## References

1. Basili, V., Caldiera, G., Rombach, H. (1994), "Experience Factory", In *Encyclopedia of Software Engineering vol. 1*, J.J. Marciniak, Ed. John Wiley Sons.
2. Agile Manifesto. <http://agilemanifesto.org>.
3. Cohen, B., Prusak, L. (2001), *Good Company: How Social Capital Makes Organizations Work*. Boston: Harvard Business Press, p. 19.
4. Kelley, R., Caplan, J. (1993, July-August), "How Bell Labs Creates Star Performers", *Harvard Business Review*, p. 128-139.
5. Haas, R., Aulbur, W., Thakar, S. (2000), Enabling Communities of Practice at EADS Airbus, in M.S. Ackerman, V. Pipek & V. Wulf, eds, "Sharing Expertise: Beyond Knowledge Management", MIT Press, Cambridge MA.
6. Lave, J., Wenger, E. (1991), *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.
7. Wenger, E., McDermott, R., Snyder, W. (2002), *Cultivating Communities of Practice*, Harvard Business School Press, Boston, MA.
8. Trittmann, R. (2001), "The Organic and the Mechanistic Form of Managing Knowledge", in K.-D. Althoff, R.L. Feldmann, W. Muller, eds, *Proceedings of the 3<sup>rd</sup> International Workshop on Learning Software Organizations 2001*, Springer-Verlag, Berlin Heidelberg
9. Greenburg, S., Roseman, M. (1998), Using a Room Metaphor to Ease Transitions in Groupware, in M.S. Ackerman, V. Pipek & V. Wulf, eds, "Sharing Expertise: Beyond Knowledge Management", MIT Press, Cambridge MA.
10. Bernstein, A. (2003), How Can Cooperative Work Tools Support Dynamic group Processes? Bridging the Specificity Frontier, in T. Malone, K. Crowston & G. Herman, eds, "Organizing Business Knowledge", MIT Press, Cambridge, MA.
11. Beedle, M., Schwaber, K. (2001), *Agile Software Development with SCRUM*, Prentice Hall, Englewood Cliffs, NJ.
12. Cunningham, W. Leuf, B. (2001), *The Wiki Way Quick Collaboration on the Web*, Addison Wesley, Reading, MA.
13. Ehrlich, K. (2000), Locating Expertise: Design Issues for an Expertise Locator System, in M.S. Ackerman, V. Pipek & V. Wulf, eds, "Sharing Expertise: Beyond Knowledge Management", MIT Press, Cambridge MA.
14. VersionOne <http://www.versionone.net> (Last Visited: September 25, 2003)
15. Xplanner <http://www.xplanner.org> (Last Visited: September 25, 2003)
16. TWiki <http://www.twiki.org> (Last Visited: September 25, 2003)
17. Henninger, S., Ivaturi, A., Nuli, K., Thirunavukkaras, A. (2002), "Supporting Adaptable Methodologies to Meet Evolving Project Needs", in D. Wells, L. Williams, eds, *Proceedings of XP/Agile Universe 2002*, Springer, Berlin Heidelberg New York.
18. Nick, M., Althoff, K.-D. (2001), "Engineering Experience Base Maintenance Knowledge", in K.-D. Althoff, R.L. Feldmann, W. Muller, eds, *Proceedings of the 3<sup>rd</sup> International Workshop on Learning Software Organizations 2001*, Springer-Verlag, Berlin Heidelberg
19. Lindvall, M., Rus, I., Sinha, S.S. (2002), "Technology Support for Knowledge Management", in S. Henninger, F. Maurer, eds, *Proceedings of the 4<sup>th</sup> International Workshop on Learning Software Organizations 2002*, Springer-Verlag, Berlin Heidelberg
20. Mahe, S., Rieu, C. (1997), "Towards a Pull-Approach of KM for Improving Enterprise in Small and Medium Enterprises", in *Proceedings of the International Symposium on Management of Industrial and Corporate Knowledge*, Compiègne.