An Agile Methodology For Implementing Knowledge Management Systems : A Case Study In Component-Based Software Engineering

Mostefai Mohammed Amine [1], Mohamed Ahmed-Nacer [2]

[1] *Ecole Supérieure d'Informatique Oued Smar, Algiers, Algeria*
[2] *Université Sciences et Technologie, Houari Boumediene Bab Ezzouar, Algiers, Algeria*
[1] *m_mostefai@esi.dz,* [2] *anacer@mail.cerist.dz*

Abstract

Component-Based Software Engineering (CBSE) is a knowledge-intensive discipline where all activities require the use and the transfer of knowledge between collaborators. A better usage, transfer and application of this knowledge are required factors to face the challenges of CBSE. In this perspective, Knowledge Management (KM) aims at setting up knowledge-focused processes in the organization. Knowledge Management Systems (KMS) are the technological infrastructure supporting these processes. In this paper, we present a new methodology of KMS implementation in a CBSE-oriented organization. We also present a case study of applying this methodology in an existing CBSE organization.

Keywords: *Component-Based Software Engineering, Knowledge Management, Knowledge Management System*

1. Introduction

Component-based software engineering (CBSE) is a discipline that consists in the development of systems as assemblies of parts (software components), the development of parts as reusable entities and the maintenance and upgrading of systems by customizing and replacing such parts [1].

A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third-parties [2].

CBSE involves many people working in different phases and activities. The knowledge used and produced in these activities is various, growing and hardly traceable: organizations have problems keeping track of what this knowledge is, where it is, and who owns it [3]. Knowledge Management (KM) provides mechanisms to create, share, transfer and apply knowledge. KM relies on a technological infrastructure called Knowledge Management Systems (KMS) [4].

This paper concerns the implementation of KMS in an organization using CBSE. To implement the KMS, we will use a new methodology that combines aspects from agile methodologies and existing KMS-oriented methodologies.

To test our methodology, we will apply it to the implementation of a KMS in an existing organization that makes an intensive usage of CBSE.

We also present in this paper the results of the application of that methodology in the organization. These results are categorized in two aspects: the static aspects which is an ontology that represent the domain of CBSE and the dynamic aspect which is the deployed KMS.

This paper is structured as follows: Section 2 concerns the definitions of "Knowledge", "Knowledge Management" and "Knowledge Management Systems". This section is also about some aspects that should be taken into consideration when implementing KMS such as the expected features of KMS and the existing methodologies. Section 3 addresses the motivations of implementing KMS in CBSE and lists some similar experiences. Section 4 details our methodology and the factors that let us propose it instead of using an existing one. Section 5 presents a case study of an application of our methodology and the related results. Finally, section 6 concludes.

2. Knowledge, Knowledge Management and Knowledge Management Systems

2.1 Knowledge

Knowledge is defined as a justified belief that increases an entity's capacity for effective action [5]. There are two types of knowledge: (1) the explicit knowledge which is a formal or semi-formal knowledge in the form of documents, manuals, formulas, etc., and (2) tacit knowledge which is a deeply individual knowledge. While the first type of knowledge is easy to share and transmit, the second type is difficult to formalize and to transmit.

Knowledge is subject to four possible transformations: (1) the externalization concerns the formalization mechanisms that allow transforming individual knowledge into sharable knowledge, (2) the internalization process consists of learning by transforming a formal knowledge to a personal knowledge, (3) the combination process is about composing multiple formal knowledge sources to create a new formal knowledge or experience and (4) the socialization process consists in combining informal knowledge without passing by any formalization.

The challenge of KM is to maximize externalization to increase the organization's capacity of using corporate knowledge.

2.2 Knowledge Management and Knowledge Management Systems

Knowledge management is a method that simplifies the process of sharing, distributing, creating, capturing and understanding of a company's knowledge [6].

The adoption of KM in professional organizations is due to many factors such as the competitiveness of the market, the constant need of innovation, the reduction of teams, the reduction of learning time, the importance of the tacit knowledge of key collaborators and the damages caused by departures [7].

Knowledge Management Systems are a class of information systems designed for sharing and integrating knowledge [8]. KMS are the technological infrastructure that makes the KM concrete in the organization.

2.3 KMS Implementation Considerations

2.3.1 KMS Main Features

The main features of a KMS are categorized in six main categories [9]: (1) the content repository feature allows the storage and the retrieval of knowledge, (2) the collaboration feature is about the communication tools permitting to link users, (3) the knowledge dissemination feature concerns the mechanisms of a quick and an efficient access to knowledge, (4) the content integration feature concerns importing knowledge from external sources, (5) the domain ontology feature targets the representation of the domain knowledge and (6) the knowledge security feature is how the KMS protects knowledge and secures the access to it.

2.3.2 KMS Implementation Methodologies

KMS constitute a special kind of information systems that are implemented using special dedicated methodologies.

There are many attempts to propose a KMS implementation methodology in literature. [10] lists many methodologies coming from both academic and industrial world and proposes its own methodology composed of twelve (12) steps. [11] and [12] are other examples of KMS implementation methodologies.

Even if the methodologies are different, heterogeneous and composed of various steps, they all include three major phases: analysis, implementation and exploitation. Each phase will be mapped into one or more steps in each methodology and will be executed in a sequential, iterative or parallel way according to the methodology.

The analysis phase concerns the extraction of the concepts and their relationships according to the target domain. This will result in an ontology and the business requirements of the KMS. The implementation phase consists in transforming the results of the previous phase into a working IT system. The exploitation phase permits to move the KMS from a development / test status to an exploitation status where users can use the KMS to share and access knowledge.

3. Motivations of Implementing KM in CBSE and Related Work

The main purpose of CBSE is to reduce time-to-market delays by using (or reusing) components. However, CBSE can also engender some risks [1, 13]: an increased time and effort required for the development of components, the consequences of bad component selection, the dependence to the component supplier and the costs of the maintenance of components.

These risks can be avoided or at least reduced if the knowledge existing in the brain of collaborators, in the corporate memory or outside the organization is effectively captured and used. The motivations of using Knowledge Management (KM) in CBSE are justified by the (1) desire of capturing and using process and product knowledge, (2) the identification of knowledge holders, (3) the integration of external or internal knowledge sources and (4) the need of avoiding making the same errors in development projects using or producing components.

The implementation of KM does not intend to be a replacement of the current software processes and tools in the organization. It is a support tool that targets making knowledge accessible, shared, easily transmissible and effectively applicable.

The community of software engineering has shown a growing interest in implementing KM in the last decade [14]. One of the most famous achievements in implementing KM in software engineering (SE) is the experience factory brought by the Basili et al. in [15]. However, the experience factory is based on a proprietary methodology called QIM (Quality Improvement Paradigm) and addresses software engineering in general and not especially CBSE.

In [16], authors detail some case studies of implementing KM in some software companies. The study focused more on both the managerial aspects and the obtained results than the technological infrastructure. Again, there is no particular focus on CBSE in this study.

4. Methodology for KMS Implementation

4.1 Motivations for a New Methodology

Many factors urged us to propose our own methodology instead of using an existing one.

The first factor is that a project conducted using existing methodologies is a long and fastidious process that requires an important investment in time, budget and resources. The second factor is that the existing methodologies do not focus on client collaboration, which can lead to the project failure [17]. For our project, the target organization which is in IT domain aimed for a tight implication in all the phases of the project.

The last factor is that our team masters and applies agile methodologies. So does the target organization. Consequently, the team desired to fructify this experience instead of learning a new one.

These factors pushed us to propose our own methodology based on agile methodologies [18] which have the following characteristics: (1) short iterations, (2) intensive customer collaboration, (3) continuous assessment and integration and (4) early identification of risks and misconceptions. In addition to “pure” agile aspects, our methodology handles the KM-oriented phases listed in the previous section which are the analysis, implementation and exploitation.

To validate our methodology, we put it directly to practice by conducting our first project targeting a CBSE organization.

4.2 KMS Implementation Methodology Details

Our methodology is an iterative process composed of a starting phase called initialization and five other iterative phases: Domain Mapping, Profile and Policies Identification, Implementation and Personalization and finally, the Validation phase. The figure 1 illustrates this methodology.

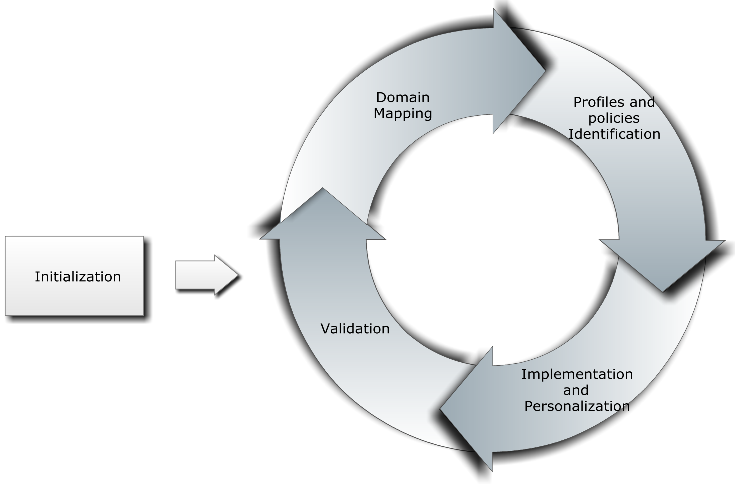
Initialization
Domain Mapping
Profiles and policies Identification
Implementation and Personalization
Validation

Fig 1 - The KMS Implementation Methodology

4.2.1 The initialization phase

This is longest and the most important phase of the methodology. This phase is about the first contact between the development team, the customer and the business domain.

The main objective of this phase is to have the deepest understanding possible of the organization, the corporate knowledge, the knowledge flows and the business concepts. It is also about understanding the culture of the organization and its environment.

There are two ways of implementing KMS: build it from scratch or by extending an existing system (this system is called base system). The second approach is generally privileged for two reasons: most of companies already have a base system installed and the second reason is that this base system includes natively some KMS features such as storage or collaboration. The base system can be a collaborative portal such as Sharepoint [19], Lotus Notes [20], an ECM (Enterprise Content Management System) such as Alfresco [21] or Documentum [22] or a DMS (Document Management System) such as Knowledge Tree [23]. The cited examples are not an exhaustive list but represent the most popular products in the domain.

The initialization phase is also about the creation of an initial ontology [24] of the domain and the selection of the main sources that will provide the concepts for that ontology. The ontology sources include the enterprise business domain, the enterprise culture, related books and manuals and also existing related ontologies.

The choices concern also the tools that will be used to create and maintain the ontology (for example Protégé [25]) and the language used to represent it (for example UML, XML or OWL [26]).

Because of the sensitive nature of that phase, it will likely be the longest phase and will last from three to six months according to the size of the organization and the complexity of the domain.

4.2.2 The domain mapping phase

The ontology created in the initialization phase is only a start for an iterative process where this ontology evolves over time. Through a continuous contact with the customers and the end-users, the domain mapping phase handles the capture of new concepts and relationships between concepts in the business domain.

In this phase, the ontology is mapped into the system. This mapping is generally performed using content types, fields and instances of content types.

4.2.3 Profiles and Policies Identification Phase

As any classical information system, KMS are used by multiple users and groups of users. Every user, according to his membership, has a set of authorized and unauthorized actions that he can perform.

The Profile and policies phase defines the authentification mechanisms and the authorization scopes of the KMS users.

4.2.4 The Implementation and Personalization Phase

In this phase, the modules allowing the usage, the capture and the processing of the new concepts are built.

The modules are developed using the adequate programming tools and languages and are deployed into test environments to assess their quality.

During this phase also, content gateways are put in place. Content gateways are special software services that are responsible of integrating external content from different sources of data. For each data source, a gateway is created that is responsible of converting provided data into elements usable by the KMS.

While the previous phase (Profiles and Policies) is about security and authorization, personalization concerns how knowledge is presented to users. For example, non-technical users will be bothered by displaying too much technical information when they try to access the KMS.

Personalization consists in developing various representation of the same knowledge according to the end-user profile.

4.2.5 The Validation Phase

In this phase, the KMS is not yet exploitable but is still usable and accessible in test servers.

In this step, KMS developers and testers ensures that the KMS is performing the expected operations and is conform to the enterprise requirements. In this phase also, access is given to a group of end users (or beta testers) that will give their feedback and acceptance about the usability of the KMS.

This phase should emphasize on the non-regression of the KMS caused by the evolution of the ontology.

5. Implementing the KMS in a Context of CBSE

5.1 The Context

Our project is a part of a larger project initiated in 2005 in our software engineering department. The project is about implementing a KMS in an enterprise that makes an intensive usage of CBSE.

The target organization is a partner of a world leader in smart cards technology and performs all her developments using .NET technologies (C#, Visual Studio, etc.). The enterprise belongs to an industrial group that focuses essentially on security. The industrial group is composed of many companies with dozens of employees each. The technical department is composed of about 30 collaborators.

5.2 The Implementation

The initialization phase was essentially about gathering the maximum information about the organization, its culture, CBSE and its challenges.

We had many sources to build the initial ontology: the SWEBOK reference [27], some ontology projects based on SWEBOK such as [28] and of course, the client experience, culture and point of view.

The choices were not difficult to take because the enterprise already had Microsoft Sharepoint 2007 in place. Our project was structured to fit the extension of Sharepoint to perform the KMS features. Our choices were consequently led to Visual Studio 2008 [29] and Microsoft Sharepoint Designer [30] as development tools and UML (Unified Modeling Language) for representing the business ontology.

After the initialization phase, we drove the process in five iterations of six weeks each. Because of the short delay, we collected quick feedback from users and around the second iteration, the development team and the end-users were speaking the same language because we eliminated many misconceptions.

The domain ontology that we created evolved to about 250 concepts. All the concepts inherit directly or indirectly from three parent concepts: Agent, Activity and Artefact.

For example, the concept Project derives from ComposedActivity that derives itself from the Activity concept. The concept Component derives indirectly from the “Artefact” concept whereas the concept “ComponentSelection” derives indirectly from the “Activity” concept.

The end users dived very quickly into the KMS because of the progressive deliveries engendered by the short iterations. The KMS serves as a central point that captures developer experiences and facilitates usage through component integration and development.

There was no rejection from the users because of their tight participation in all the stages of the project.

Table 1 summarizes the execution and the overall scenario of the project. Each line provides a description of the results obtained in the related phase.

Table 1 – KMS Implementation Phases

Iteration	Duration	Phase	Results
0	3 Months	Initialization	Sharepoint as a base system, VS 2008 and Sharepoint Designer as tools, UML to describe ontology and business concepts, First contact with the organization, about 50 concepts for the initial ontology. SWEBOK, CBSE resources (books, white papers) and the enterprise culture as the ontology sources.
1	6 Weeks	Domain Mapping	First concepts about CBSE, hierarchical parents : artifact, agent and activity, about 80 concepts
		Profiles and Policies	Identification of technical and non-technical profiles
		Implementation	Implementation of the first forms for acquiring knowledge about projects, components and collaborators
		Validation	Testing the modules implemented, end-users feedback
2	6 Weeks	Domain Mapping	Activities, projects and development methodologies
		Profiles and Policies	Additional profiles : designers, architects, developers, testers
		Implementation	Forms for additional concepts, screens and forms concerning activities, related validation rules
		Validation	Scenarios according to the ontology evolution
3	6 Weeks	Domain Mapping	Modeling software tools, component-oriented concepts such as Selection and Identification, about 140 concepts
		Profiles and Policies	Component-oriented profiles such as component selectors, security rules for collaborators
		Implementation	Screens and forms for new concepts
		Validation	Simulation projects and non-regression tests (tests of the previous iterations
4	6 Weeks	Domain Mapping	About 220 concepts in the ontology, new concepts about external elements: suppliers, customers, organizations

		Profiles and Policies	Map security with Active Directory
		Implementation	Collaboration forums and chats, portals, wikis
		Validation	Non-regression tests, Acceptance tests
5	6 Weeks	Domain Mapping	About 250 concepts
		Profiles and Policies	Finalizing integration with active directory, non-technical profiles personalization and filters
		Implementation	Integration with external sources (code repository, file servers), global deployment
		Validation	Global acceptance tests

5.2.1 Results

5.2.1.1 The Ontology

The ontology was progressively constructed basing on the collaboration of the target organization and SWEBOK.

We used UML as a formalism to represent the ontology. Concepts were directly mapped into Sharepoint content types. The ontology contains about 250 concepts.

Figure 2 is an example of a subset of the UML diagrams representing the component selection activity.

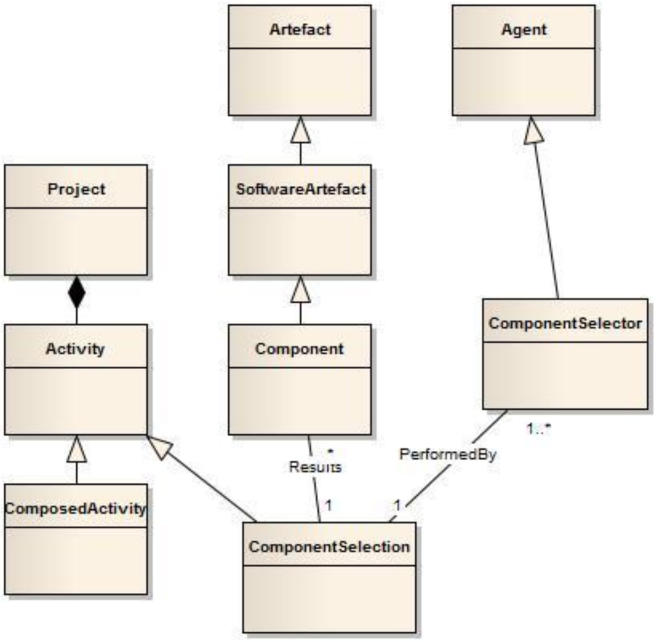

Fig 2. An UML Example of a Subset of the Ontology

This example represents the component selection activity which inherits from the activity concept. A project inherits from ComposedActivity, a special activity composed of other activities including eventually the selection.

The component selector, a descendant from the concept “Agent” is the collaborator who performs the selection.

5.2.1.2 The KMS

The architecture of our KMS is illustrated in figure 3:

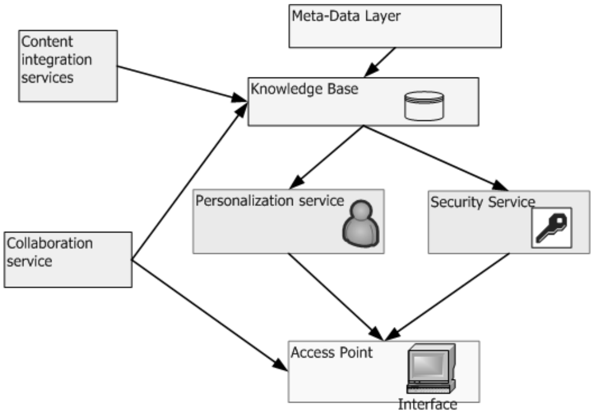

Fig 3. The KMS Architecture

Our KMS is a multi-layered system built on top of Sharepoint. The table 2 summarizes the KMS layers and how they implement the features cited in section III:

Table 2 – KMS Architectural Layers and Features

Layer	Features	Implementation	Description
Meta-Data Layer	Domain Ontology	Sharepoint content types and custom fields	This layer is the ontology representation using Sharepoint content types. It includes all the domain concepts and relationships
Knowledge Base Layer	Content Repository and Knowledge Dissemination	SQL Server and Sharepoint persistence services	This layer stores all the knowledge instances related to CBSE domain such as components, projects, activities, collaborators, etc
Content Integration Service	Content Integration	Code repositories (version management system), file system	This layer is responsible of the automatic collection of external knowledge resulting from miscellaneous sources such as files, source code repositories, document repositories, internet, etc.
Security Service	Knowledge Security	Active Directory	Authenticates collaborators and processes and define their access level and action perimeter
Personalization Service	Knowledge Dissemination	CSS, web forms, ASP.NET	Adapts (form and content) knowledge elements to the profile of the current user. For examples, filters "too technical" information to the "manager" profile
Collaboration Service	Collaboration	Sharepoint collaboration features	Provides a set of tools allowing collaborators to be synchronously or asynchronously linked (for example forums, chats, mailings…etc.). In addition to that, this service provides a gateway to the content integration service for capturing knowledge created or transferred in communications (chat captures, resolved forum queries, etc.).

Access Point	Knowledge Dissemination	Sharepoint forms and intranet features, search engines	The access point allows entering knowledge queries, creating or updating knowledge instances and evaluating existing knowledge instances. The access point is a simple internet browser that provides access to Sharepoint portal.

5.2.1.3 Results Interpretation

The major outputs of the methodology consist in two aspects: the static aspect that concerns the concepts related to CBSE and their relations and the dynamic aspect which is the KMS itself.

Although the conduction of the methodology was performed in a CBSE organization, there have been no specific concepts to that organization. The SWEBOK [27] and other CBSE-related resources provided most of the concepts of the ontology. The positive impact of the resulting ontology is that it is highly reusable in other KMS projects targeting organizations of the same domain.

Concerning the KMS itself, the extension strategy that we adopted lead us to produce an architecture that is based on the architecture of the base system. Consequently, some aspects such as security, presentation and storage depend tightly on how are they implemented in this base system (Sharepoint 2007).

In addition of permitting to the target organization to handle the KM processes such as creation, storage, transfer and retrieval, this experience can be an accelerator for other projects having one of the following similarities: implementing KMS in CBSE by using another base system (ontology reuse) or implementing KMS in another domain using the same base system (the project artefacts reuse).

6. Conclusion

CBSE is a knowledge-intensive activity where collaborators produce and consume knowledge during all the development phases. An effective usage, transfer and capture of this knowledge are vital to the survival and the competitiveness of organizations using CBSE. To ensure that, KM is the solution and the KMS are the technological infrastructure needed for the success of that solution.

In this paper, we proposed a new methodology for the implementation of KMS. This methodology is based on concepts brought both from agile methodologies and existing KMS implementation methodologies. The main objectives of our methodology are: an early elimination of risks and misconceptions by ensuring short iterations, continuous integration and intensive customer collaboration.

To be executed, our methodology requires less resources and budget than existing methodologies. To validate our methodology, we put in practice by using it in the implementation of a KMS in a CBSE organization.

The project was conducted in 42 weeks and five iterations of six weeks each. Microsoft Sharepoint was used as a base system to the implementation of the KMS. The resulting ontology contained about 250 concepts coming from SWEBOK and the customer experience.

In the first weeks of the KMS exploitation, the results were satisfying and the customer did not encounter difficulties to adopt the KMS.

Although the results are positive, some work remains to do. At the implementation level, it would be interesting to reproduce the experience in other companies using different base systems (such as Alfresco or Documentum). At a managerial level, our methodology addresses essentially explicit knowledge that is manipulated by the KMS. We have to improve the methodology to propose mechanisms to handle tacit knowledge and motivate users to produce, to share and to assess knowledge.

Finally, at a transversal level, we plan to apply the methodology to a broader domain (general software engineering) and eventually, to a categorically different domain that requires KM implementation.

References

[1] I. Crnkovic, "Component-based software engineering: New challenges in software development," CIT. Journal of computing and information technology, pp. 151-161, 2003.

[2] C. Szyperski, Component Software: Beyond Object-Oriented Programming, Second Edition ed.: Addison-Wesley, 2002.

[3] I. Rus and M. Lindvall, "Knowledge management in software engineering," IEEE Software, vol. 19, pp. 26-38, 2002.

[4] M. Alavi and D. E. Leidner, "Review : Knowledge Management and Knowledge Management Systems : Conceptual Foundation and Research Issues," MIS Quarterly, vol. 25, pp. 107-136, 2001.

[5] I. Nonaka and N. Konno, "The concept of "Ba": building a foundation for knowledge creation," California Management Review, vol. 40, pp. 40-54, 1998.

[6] T. Davenport and L. Prusak, Working Knowledge: How Organizations Manage What They Know: {Project Management Institute}, 1997.

[7] R. O. Barclay and P. C. Murray. (1997, What is knowledge management?

[8] M. Alavi, et al., "Knowledge management systems: issues, challenges, and benefits," Communications of the AIS, vol. 1, 1999.

[9] W. Staniszkis, "Feature Requirements of a Knowledge Management System," Rodan Systems S.A.2003.

[10] F. A. Calabrese and C. Y. Orlando, "Deriving a 12-step process to create and implement a comprehensive knowledge management system," VINE, vol. 36, pp. 238 - 254, 2006.

[11] H. Smuts, et al., "A framework and methodology for knowledge management system implementation," presented at the Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, Vanderbijlpark, Emfuleni, South Africa, 2009.

[12] R. Chalmeta and R. Grangel, "Methodology for the implementation of knowledge management systems," J. Am. Soc. Inf. Sci. Technol., vol. 59, pp. 742-755, 2008.

[13] J. Voas, "COTS software: the economical choice?," Software, IEEE, vol. 15, pp. 16-19, 1998.

[14] F. Bjornson and T. Dingsoyr, "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used," Information and Software Technology, vol. 50, pp. 1055-1068, 2008.

[15] V. R. Basili, et al., "THE EXPERIENCE FACTORY," Encyclopedia of Software Engineering, vol. 2, 1994.

[16] T. Dingsøyr and R. Conradi, "A survey of case studies of the use of knowledge management in software engineering," International Journal of Software Engineering and Knowledge Engineering, vol. 12, pp. 391–414, 2002.

[17] C. M. Cullough. (2005). What Is Knowledge Management - Knowing What We Know? Available: http://www.isnare.com/?aid=6711&ca=Business+Management

[18] S. Augustine, Managing Agile Projects: Prentice Hall; illustrated edition, 2005.

[19] D. Sterling, Microsoft® Office SharePoint® Server 2007: The Complete Reference, Mc Graw Hill ed., 2008.

[20] C. Bobrowski and G. Kreymann, "Knowledge Management in Internal Medicine Using Lotus Notus – A Knowledge Management Perspective," in Professional Knowledge Management. vol. 3782, K.-D. Althoff, et al., Eds., ed: Springer Berlin / Heidelberg, 2005, pp. 339-348.

[21] Alfresco. (2010). Alfresco Products and Services. Available: http://www.alfresco.com/products/

[22] EMC. (2010). EMC Documentum. Available: http://www.emc.com/domains/documentum/index.htm

[23] KnowledgeTree. (2010). KnowledgeTree Features. Available: http://www.knowledgetree.com/features

[24] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," International Journal of Human-Computer Studies, vol. 43, pp. 907-928, 1995.

[25] W. E. Grosso, et al., "Knowledge modeling at the millennium (The design and evolution of Protégé-2000)," Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Canada, 1999.

[26] M. Lenzerini, et al., "Ontologies Representation & Reasoning," Universit di Roma La Sapienza2004.

[27] A. Abran, et al., Guide to the Software Engineering Body of Knowledge - SWEBOK: IEEE Press, 2004.

[28] O. Mendes and A. Abran, "SOFTWARE ENGINEERING ONTOLOGY: A DEVELOPMENT METHODOLOGY," Metrics News, vol. 9, pp. 68-76, 2004.

[29] Microsoft. (2008, 2010). Visual Studio. Available: http://msdn.microsoft.com/en-us/vstudio/aa718325

[30] Microsoft. (2007). Microsoft Office SharePoint Designer 2007 Top 10 Benefits. Available: http://office.microsoft.com/en-us/products/microsoft-office-sharepoint-designer-2007-top-10-benefits-HA010165022.aspx?CTT=1