Know-Cap: A Method for Knowledge Capitalization in Software Engineering

Gislaine Camila Lapasini Leal[1,2], Paulo Cezar Stadzisz[1] and Elisa Hatsue M. Huzita[3]
[1]*CPGEI, Federal Technological University of Parana, Curitiba, Paraná, Brazil*
[2]*Department of Production Engineering, State University of Maringá, Maringá, Paraná, Brazil*
[3]*Department of Computing, State University of Maringá, Maringá, Paraná, Brazil*

1 STAGE OF THE RESEARCH

The research is in the design phase of the capitalization method. At the moment the steps, activities and guidelines for the capitalization of knowledge in Software Engineering were defined. It is named Know-Cap method.

The Figure 1 shows a conceptual model of the Know-Cap method. The faces of the prism represent the source for obtain knowledge (people) and the steps of the method (Identify, Add Value, Maintain and Monitor). As it can be seen in Figure 1, the people is the base of the capitalization method. The representation emphasizes that people is the base of the capitalization process, since they hold, use and generate knowledge throughout the software development. In addition, they participate in identifying the knowledge demands. The step Monitor acts as an integrator by allowing the monitoring of the evolution of knowledge, by identifying its obsolescence and/or valuation.

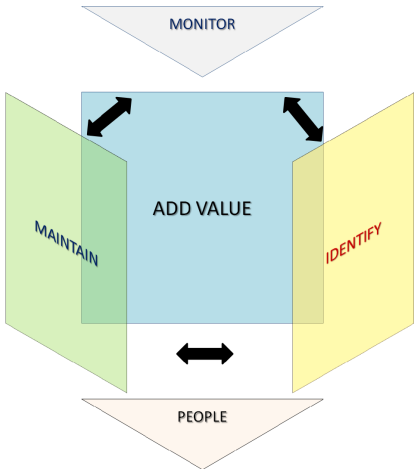

Figure 1: Know-Cap.

Also, analyzing Figure 1 it is possible to observe by means of two-way arrows, the Know-Cap method is iterative. It enables to refine the artifacts and obtain feedback in relation to the demands and the knowledge development.

The proposal of Know-Cap was guided by the following assumptions:

- Prescriptive: the method provides guidelines for the implementation of activities, highlighting what should be done and how. Were also defined templates for the generated artifacts, enabling standardization and formal way to communicate.
- Focused on the core areas of software engineering: the method highlights the conversion of knowledge into organizational assets related to Requirements, Design and Implementation stages. Independence from the development methodology: the method may be used regardless of the approach used in the development, being it traditional or agile. However, it is noteworthy that the methodology influences the amount of knowledge that is capitalized. For example, the artifacts defined by templates waterfall and RUP promote the materialization of knowledge, ie, assist in the process of outsourcing. On the other hand, the shorter iterations and continuous planning of agile methods facilitate the sharing of tacit knowledge, ie socialization.
- Independence of company size: the method can be adopted by companies of any size, regardless of project or team size.

For Know-Cap adoption should be considered the involvement of top management, definition of a team responsible for selection of a pilot project. Support from top management is essential to ensure the necessary resources to capitalize knowledge. The team responsible must involve at least one employee in each area (Requirements, Design and Implementation). Team members will be responsible for promoting the activities funded.

Camila Lapasini Leal G., Cezar Stadzisz P. and Huzita E. (2013).
Know-Cap: A Method for Knowledge Capitalization in Software Engineering.
In *Doctoral Consortium*, pages 28-36

The effectiveness of capitalization depends on organizational policies, strategies and managerial and cultural aspects, not just the method itself.

The following sections describe the steps that comprise the method.

1.1 Identify

It is a strategic step that aims to identify the demands of knowledge and its importance to aid decision making and support the organization's processes. This step should involve the largest number of participants (project manager, developers, analysts, and others), as the other directs. The activities of this step, namely: identifying knowledge gaps, analyze the knowledge demands, knowledge selection, identify the location and knowledge classifying.

Identify Gaps in knowledge it is a planning activity in which the demands of knowledge must be identified. It is important to involve all stakeholders in the process, since this activity will guide the remaining steps of the capitalization of knowledge and enable direct efforts. To identify these gaps can be used techniques, such as interviews, SWOT (Strengths, Weaknesses, Opportunities, Threats), brainstorming and benchmarking.

Analyze the Demands of Knowledge aims to map who are interested, potential benefits and availability of each knowledge identified. The sources for obtaining knowledge can be internal or external to the organization and include training, consulting, workshops and technical visits.

Select Knowledge involves identifying what are the critical knowledge, ie knowledge that has greater value for the organization. (Kokkoniemi and Harjumaa, 2009) emphasize that it is virtually impossible to collect and record all the knowledge involved in an organization, regardless of its size. Due to constraints of time and cost it is not possible to capitalize on all the knowledge involved in the activities, so must be selected the skills that enable higher returns. The knowledge selection can be made from the following elements: the experience of the project manager and team, historical project data, expert judgment or based on the requirements of the maturity model CMMI (Capability Maturity Model Integration). Other factors that may assist in the knowledge selection is the risk analysis of loss and identifying key areas of knowledge.

Identify the Location is to determine where the explicit knowledge is embodied (artifact, document, tool). Regarding tacit knowledge is necessary to identify the people in the organization who has it. Where knowledge is not available in the organization the mechanisms of skilled labor acquisition or training can be used.

Classify the Knowledge is an activity that aims to organize the knowledge and thus, facilitate their recovery, dissemination and access. The classification allows standardizing the vocabulary and, thus establishing a standard language for interaction and sharing of knowledge and experiences, which minimizes communication problems. The categories defined in Know-Cap for the knowledge classification are:

- Domain Knowledge: refers to the specific knowledge of the application domain, ie, the business rules.
- Process Knowledge: refers to knowledge about the structure of the work, including the definition of responsibilities and tasks.
- Resources Knowledge: includes knowledge about the technical resources (tools such as case tools, version control tools, project management tools, development environment) and non-technical resources (human resources) used.
- Technology Knowledge: involves knowledge related to norms, standards, approaches, programming language, database, frameworks and others.
- Technical Knowledge: refers to the final products and by products generated.
- Management Knowledge: refers to knowledge of planning, monitoring and conducting the project.

1.2 Add Value

The purpose of this step is to increase the value of knowledge and it occurs through the use, exploitation and reuse them. The step involves the following activities: explicit, appraise, disseminate, instantiate and update knowledge.

Explicit Knowledge refers to the externalization of tacit knowledge identified. The knowledge conversion can be performed by interviews, observation, brainstorming, and others. These techniques constitute the first step for the externalization. They are informal approaches in which the terms are not precisely defined. As a way to support knowledge explicitness can be used the following practices:

- Define a glossary of terms related to the application domain.
- Formalize the good practices of coding, identifying the language to be adopted during implementation, variable naming, functions naming, constant naming, comments, indentation and others. It is a knowledge that facilitates the integration of new team members.

- Document lessons learned, keeping track of what has occurred and an analysis of causes. The lessons learned can be identified in each of the steps (Requirements, Design and Implementation) and shall include all types of experiences, not just the successful results.
- Develop rationale recording founded problems, decisions made, alternatives considered, criteria and arguments that led to the decision. (Dutoit and Paech, 2000) and (Gueraich and Boufada, 2011) indicate that this type of initiative introduces an initial workload. However, it is beneficial to support decision making in the later stages, usually, as well as new projects.
- Document change requests. It is an activity that make possible track changes and maintaining traceability, ie, monitoring the generated artifacts.
- Prepare documents for inspection. It is an activity connected to each stage of development that allows to acquire and reuse knowledge (Kokkoniemi, 2006).

Value the Knowledge consists in measuring the value of each asset of knowledge. It is noteworthy that the value of knowledge is not static, it evolves with time, circumstances and priorities. That is, the knowledge value can be increased (increase) or decremented (depreciation) over time. To establish the knowledge value can be used techniques of activity-based costing method or cost centers.

Disseminate the Knowledge involves defining who are interested in every kind of knowledge and establish appropriate mechanisms for knowledge distribution. Strategies for dissemination of knowledge must consider the several perspectives on the importance of each knowledge category (Figure 2). This perception impacts the extent to which knowledge is required and also the perception of its value.

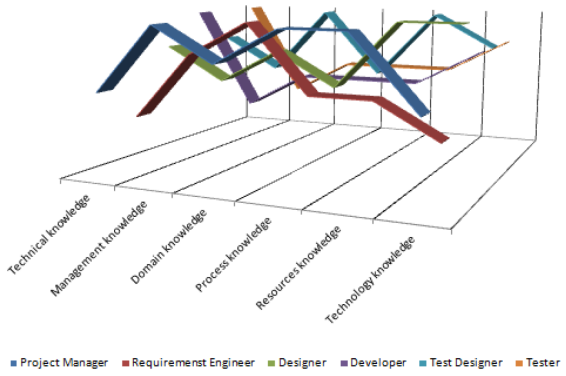

Figure 2: Knowledge category.

Instantiate the Knowledge is to apply the knowledge in problem solving and decision making. The knowledge reuse is an activity that adds value, ie monetize knowledge. It is the activity in which the return on investment is generated. The reuse of knowledge allows to avoid rework and supports in solving recurring problems, which allows to obtain improvements in the quality and productivity.

Update Knowledge involves the modification of knowledge from experience. Aims at the maintenance of knowledge, that is, keep them updated to allow the reuse.

1.3 Maintain

It is a step that aims to preserve the knowledge acquired. This stage involves the following activities: represent and store of knowledge.

Representing involves modeling formally the knowledge to enable effective understanding, communication and facilitate the reuse. The formal representation of knowledge can be performed using the following techniques: logic, taxonomy, ontology, frames, semantic networks and other

Store is the activity that allows retain knowledge so that it can be distributed and reused.

1.4 Monitor

It is a step that aims to establish continuous coordination among the remaining steps, evaluate the results, monitor the dynamics of knowledge and provide feedback for refinement of crucial knowledge. The metrics permit to quantify the objectives and reflect in the performance. The Know-Cap presents an initial set of metrics that allow monitoring, evaluation and understanding of the corporation knowledge.

2 OUTLINE OF OBJECTIVES

The main goal is to specify a method for capitalize knowledge in software engineering. The specific goals are to:

- Identify the categories and knowledge involved in Software Engineering;
- Define strategies to select the crucial knowledge;
- Define strategies to value the crucial knowledge;
- Specify the method of capitalization of knowledge, identifying their steps, guidelines and techniques that can be used;
- Representing the method appropriately to enable effective communication and understanding its;

- Evaluate the proposed method by conducting multiple case studies and experimental studies;
- Refine the proposed method.

3 RESEARCH PROBLEM

The software production is a knowledge-intensive activity, where prevail the cognitive activities. So, knowledge is an important factor for production. The main assets involved in software development is the intellectual capital, ie, the knowledge greatest source of value creation. In addition, the following factors are in features and challenges inherent to software development: i)the software is an intangible, which costs are concentrated in engineering. It is a high value asset that multiplies without generating new costs; ii) the software does not suffer physical depreciation, possible failures are the result of faults of project; iii) diversity and volume of knowledge involved, which may be related to processes, products and skills; and, iv) dynamics of evolution technologies, techniques and methodologies.

Software production is dependent on knowledge by those involved, which take various decisions, each with several options available. In this sense, the experience constitutes a valuable resource that can be exploited ((Kavitha and Ahmed, 2011); (Panagiotou and Mentzas, 2011); (Franca et al., 2012).Another aspect to be considered in the production of software refers to the dynamics of knowledge that evolves along with technology, culture and practices adopted (Iuliana, 2009).

Software Engineering, over the years has supported the software production with theories, techniques and methods. However, the increased complexity of software projects, scope and smaller delivery time have confirmed that projects often go beyond the schedule and estimated costs and therefore do not meet the quality requirements specified by the customer (Dingsyr, 2002).

Other factors that also slow down productivity and increase production costs are related to staff turnover and difficulty in identifying, locating and using knowledge. By integrating new team members, usually, there is a need to develop skills and competencies related to the processes, technologies and tools used, as well as knowledge about the domain. The above factors, in most cases, result in the inability to meet demand, resulting in economic loose and threats to competitiveness. The knowledge capitalization shows itself as an effective alternative to address these issues, it aims to convert knowledge into organizational assets so that they can be managed and thus minimize the problems related to: loss of knowledge, learning curve, repeating mistakes , rework and staff turnover.

Knowledge Management shows up as an alternative to improve the efficiency in software engineering, since it make possible to capture, disseminate, reuse the generated knowledge. So, it is make possible to obtain better quality product and also increase team productivity (Vasumathy, 2012). Furthermore, (Komi-Sirvio et al., 2002) and (Chongsringam and Prompoon, 2008) show that knowledge management supports process improvement and its products. Therefore, to achieve the requirements of time, cost and quality, organizations need to define ways to manage adequately the range of skills involved in Software Engineering.

Knowledge Management in Software Engineering has been addressed by several authors who suggest that Knowledge Management can be used to reduce development time, improve decision making, promote good practices, facilitate communication and human resource allocation, and so, improve estimates, avoid repeat mistakes, provide cheaper products, among others.

Models, practices, techniques and tools for knowledge Management in Software Engineering are highlighted in the literature. In general, practical, techniques and tools are presented as a solution to a specific problem, such as knowledge acquisition. However the models, due its abstraction level, define what should be done for Knowledge Management, provide guidelines for the instantiation of it. In this context, a gap and a research opportunity to define a method that allows to convert Software Engineering knowledge in organizational assets. This conversion from artifacts, which represent the knowledge materialization, since they are designed to meet the strategic objectives and represent investments which, it is hoped, that add value (SEI, 2010).

The knowledge capitalization includes aspects related to the location, preservation, value addition and updating of knowledge, in order to use it in the implementation of new tasks and, thus, increase the company's capital. In the context of Software Engineering, the knowledge capitalization facilitates the access, minimize the loss of knowledge, reduce the learning curve, avoid repetition mistakes and rework. So, a capitalization method is presented as a solution to facilitate access and to reduce the loss of critical knowledge, which impacts on reducing costs and development time, as well as to improve the quality of software products.

4 STATE OF THE ART

The knowledge capitalization is an important step in the process of organizational knowledge generation. It aims to reuse the knowledge, previously stored and modeled, in order to perform new tasks((Simon, 1996); (Morello et al., 2005); (Butdee, 2011)).

According to (Butdee, 2011), capitalization aims to build a capital information and improve it from disclosure. It is a process to use, exploit and reuse knowledge. (Rasovska et al., 2008) describe the knowledge capitalization as the formalization of experience in a specific field. (Grundstein and Rosenthal-Sabroux, 2005) suggest that the identification and evaluation of knowledge, which justifies the capitalization assumes a process of decision making.

To (Busch, 2006), the capitalization of knowledge is a mechanism to formalize the knowledge of judging the knowledge produced and used as the company's wealth and profit from it, thus helping to increase the amount of capital. According to (Sarirete and Chick, 2008) capitalization is the process to identify, locate, model, store, access, use, reuse, share and update knowledge. In this same sense, (Rasovska et al., 2008) show that the main purpose of capitalization is to locate, explain, maintain, access, use, update and disseminate knowledge in order to value it.

According to (Bolanle et al., 2009) the knowledge capitalization can be seen as the task of mapping existing knowledge specifying what, when and where in order to reuse it.

(Matta et al., 2001) define four steps for knowledge capitalization , namely: i) extraction and formalization of knowledge ii) knowledge sharing; iii) reuse and appropriation of knowledge, and, iv) development of organizational memory.

(Grundstein, 2000) emphasizes that the capitalization is not just a technical activity, but an essential management activity. (Tseng and Huang, 2005) show that it is necessary to identify the fundamental knowledge for the organization. This identification is to define, locate, characterize and classify the knowledge to be capitalized. In addition, the authors emphasize that it is important to capitalize, especially tacit knowledge.

(Grundstein et al., 2003) present an approach to solving problems of knowledge capitalization, characterized by four facets (locate, preserve, add value and update) and their interactions (manage).

The first aspect refers to the location of the critical knowledge, which is knowledge (explicit knowledge) and know-how (tacit knowledge) necessary for decision making and for the progress of the key processes that constitute the core of activities. It is necessary to identify them, to find them, describe them, estimating their economic value and organize them hierarchically. (Tseng and Huang, 2005) emphasize that the choice of knowledge to be capitalized, often, is conditioned by the availability of tools and / or processes, without really considering the question of the utilization of such knowledge, ie, identify the problems that need knowledge.

(Rasovska et al., 2008) mention that the process of identification and location of knowledge depends on the objectives and requirements of knowledge management. In addition, the authors highlight that the knowledge acquisition is performed from a domain analysis, technical documents and interviews with experts.

The second aspect is related to the preservation of knowledge and know-how, which encompasses the activities of acquiring, modeling, formalizing and preserving knowledge. The third aspect concerns the added value is therefore necessary to increase the value of knowledge, putting it at the service of development and expansion of the company. Ie, knowledge must be accessible in accordance with rules of confidentiality and security, disclosed, shared and used effectively to be able to be combined and generate new knowledge. According to (Rasovska et al., 2008), the third facet is the capitalization of knowledge, ie, make knowledge accessible to integrate it and spread it.

The fourth aspect concerns the updating of knowledge and know-how, which includes assessment activities, updates, standardization, enrichment of knowledge according to the experiences and new knowledge creation. (Rasovska et al., 2008) emphasize that this step is based on feedback and experience.

The fifth facet, management, is related to the interaction among the facets mentioned above and covers all management actions in order to respond to the problem of capitalization of knowledge, namely: aligning knowledge management with the strategic directions of the organization, raising awareness people; training, encourage and motivate all stakeholders of the organization; waking up the implementation of favorable conditions for cooperative work and encourage knowledge sharing; developing indicators for monitoring and ensuring the coordination of actions to measure results and determine the relevance and impact of the actions.

(Renaud et al., 2004) and (Marcandella et al., 2009) highlight that the main steps for knowledge capitalization are: locate (identify and characterize), preserve (model and store), value (use, access, dissemination and creation) and maintain (update and improve). According to (Marcandella et al., 2009) the

knowledge capitalization can be described as a cycle that involves data collection, data selection, data validation and knowledge modeling .

(Grundstein, 2000), (Grundstein et al., 2003) and (Grundstein and Rosenthal-Sabroux, 2005) propose a framework GAMETH, which provides information leading to the identification of problems, clarification of the knowledge requirements, and it also, identification, location, specification and evaluation based on the value to determine the crucial knowledge.

The main steps of the GAMETH framework are: framing the project, identification of crucial knowledge and determination of the axes of knowledge management initiatives. The first step is to specify the context of the project, the domain, the limits of intervention and the processes that will be analyzed, ie, the processes sensitive. The second step identifies the activities that may pose risks to sensitive processes. This identification is performed through the following steps: modeling of sensitive processes; determination of the critical activities of sensitive processes, identification of constraints, and outline the potential crucial knowledge. Finally, in the third step are defined, identified and characterized the knowledge to be capitalized. This step consists of the following activities: clarification of the knowledge requirements, location and characterization of knowledge, assessing the value of knowledge and determination of critical knowledge; outline of the project to improve the processes of decision making, and determination of the axes of knowledge management initiatives.

Case studies with the framework GAMETH in French Institue of Petroleum (IFP), PSA Peugeot Citroen and the French National Center for Scientific Research (CNRS) which enabled to show the relevance of the framework to clarify the requirements of knowledge.

(Matta et al., 2001) describe how to capitalize the knowledge using the MASK method (Method of Analysis and Knowledge Structuring). The authors show that this method can be used in several fields such as: security, business processes, mechanical design and others. With MASK method the knowledge is structured in systemic analysis ergocognitive, psicocognitive, and historical evolution.

(Sarirete and Chick, 2008) present a model to solve the problem of knowledge capitalization, tacit and explicit ones, in the field of engineering, within an online community of practice. The knowledge capitalization process is proposed to locate the critical knowledge (identification, mapping and classification), update it, improve it and preserve it (modeling, formalizing and archiving), bringing several perspectives of community members in different contexts.

In (Sarirete and Chick, 2008) knowledge representation is made through ontologies, in which knowledge is categorized as experiential, conceptual, systemic and routines ones. The experiential knowledge consists of practical experiences, skills acquired through discussion, dialogue and common practices. Conceptual knowledge is explicit knowledge articulated through images, symbols and languages. The systemic knowledge consists of product specifications, manuals and documents. Finally, routines knowledge is the tacit ones that is customized and embedded in actions and practices.

The capitalization process model was tested in two online communities of practice, which share their documents in a repository for easy access. The study results indicated that the majority of knowledge was capitalized as systemic ones. The experiential knowledge, conceptual and routine are not capitalized.

(Rasovska et al., 2008) present an approach for knowledge capitalization in maintenance. The approach consists of four steps: detect, protect, update and capitalize. The detection Phase detection is used analytical methods and tools of maintenance engineering. In preservation step are used UML (Unified Modeling Language) diagrams, particularly the class diagram for representing knowledge. Case-based reasoning is the technique to facilitate the knowledge reuse. The authors mention that the proposed approach suggests the implementation of an information system that includes such steps and automate the process of knowledge capitalization.

(Rodriguez-Rocha et al., 2009) describe an ontology for the knowledge capitalization in the automotive industry. The ontology is described based on ISO / TS 16949 and enables knowledge representation, manipulation and retrieval of documents.

(Butdee, 2010) and (Butdee, 2011) present a model of knowledge capitalization to design injection molding. The model is divided into three parts: knowledge capitalization, knowledge-based system and products and requirements. The knowledge-based system is integrated with the capitalization to explore, reformat and reuse knowledge. It used the case-based reasoning technique. The representation of the case is performed by means of global and local problems. In cases of recovery nearest neighbor technique is used. The authors emphasize that the most important part for capitalization is the package of organizational memory, which consists of a dynamic memory.

(Castillo-Barrera et al., 2011) use ontologies and semantic technologies to capitalize the knowledge in a factory-based software components. The authors note that it is possible to capitalize the knowledge us-

ing ontologies, because they have a more significant meaning. Furthermore, the use of ontologies enables the search for information about a specific component using intelligent techniques, such as production rules. The paper only presents information on the capitalization is performed, only highlights the possibility of capitalization.

So, Were not found in the literature, to date, studies about the knowledge capitalization in software engineering. However there are reports on the knowledge capitalization in other areas, as described above. Not being presented papers that address how to treat capitalization in knowledge intensive activities like software engineering.

Furthermore ((Grundstein, 2000), (Grundstein et al., 2003), (Grundstein and Rosenthal-Sabroux, 2005), (Sarirete and Chick, 2008), (Rasovska et al., 2008), (Butdee, 2010), and (Butdee, 2011)) describe models / frameworks for the knowledge capitalization. These studies focused on the steps involved in the cycle of capitalization. However, they not provide information on how to conduct each one of these steps, ie, find the critical knowledge, how to preserve knowledge, add value and update it. The level of detail of the models does not identify how each of these steps must be conducted so that the knowledge to be capitalized.

The framework proposed by ((Grundstein, 2000), (Grundstein et al., 2003), (Grundstein and Rosenthal-Sabroux, 2005)) is generic, ie it can be applied in any industry. Case studies were conducted using this framework in the automotive industry and research. The model presented by (Sarirete and Chick, 2008) is facing the engineering domain. The approach of (Rasovska et al., 2008) is focused on maintaining and the model proposed by (Butdee, 2010) focuses on capitalizing on the design of injection molding.

(Matta et al., 2001), (Rodriguez-Rocha et al., 2009) e (Castillo-Barrera et al., 2011) present works that describe techniques that can be used in the knowledge capitalization. These studies do not provide information about the steps of capitalization.

In (Rodriguez-Rocha et al., 2009) and (Castillo-Barrera et al., 2011) were used ontologies to capitalize the knowledge. (Rodriguez-Rocha et al., 2009) present an ontology for the automotive sector and (Castillo-Barrera et al., 2011) an ontology for a plant-based software components. (Matta et al., 2001) describe how the MASK method can be used to capitalize knowledge.

It is possible to notice that the works can be grouped into two fronts. One that describes models for capitalization and one which highlights techniques (ontologies, semantic technologies, and case-based reasoning) that can be employed to capitalize knowledge.

In general, the studies show that models follow the same structure as capitalization, which is to identify critical knowledge, maintain, capitalize and update. These models are generic, not specific to a domain, and are not prescriptive, ie, do not provide information on how to perform each of these steps, thus setting a gap and a research opportunity.

In relation to software engineering, in the literature, aspects related to Knowledge Management are discussed. However, capitalization is still a topic to be explored. In this regard, it is noteworthy that the gap is the definition of a capitalization method specific to the area of Software Engineering, which allows to identify, select and convert knowledge into organizational assets.

5 METHODOLOGY

The research can be characterized as applied, which aims to generate knowledge for practical application.

It is planned to conduct this research in three phases: Exploratory Study, Definition and Refinement Method.

In Exploratory Study phase a literature review was performed. Whit that a consistent theoretical basis for continuing study and visualize the state of the art. Concepts related to software engineering, CMMI maturity model, capitalization of knowledge, knowledge management and knowledge management in software engineering were also studied.

In the second phase, the Know-Cap will be defined. It will be performed based on the gaps identified on the previous phase. So, the main activities for the method and also to adequate representation for the knowledge will be defined.

After the method definition, will occur the Refinement phase. Initially is expected to undertake the following activities:

- Feasibility Study: the aim is to acquire knowledge about the application of the method, allowing the generation of new hypotheses. This study will be conducted in the academic and industrial community.
- Case Study: the objective it to determine the practical feasibility of the method in order to improve the understanding of researchers and characterize the application.

6 EXPECTED OUTCOME

The main result is a method for knowledge capitalization that enables to convert knowledge into organizational asset. It can also highlighted the following results: the knowledge categorization in software engineering, which facilitates access and retrieval, to identify strategies that enable to select the critical knowledge, and the definition of guidelines that guide the implementation of activities, as well as the definition of templates for each of the artifacts.

A capitalization method provides guidelines regarding the identification and selection of the critical knowledge, which enables to obtain improvements on the software development process and also on product quality. The improvements are related to integration of new member since they will have access to the organizational knowledge. So, using Know-Cap, the idea is to avoid knowledge loss, reduce learning curve and improve knowledge retention.

REFERENCES

Bolanle, O., Amos, D., and Adenike, O. (2009). Representation of knowledge resource in the context of economic intelligence systems. In *IX Congress ISKO*.

Busch, H. (2006). *Towards the re-use of knowledge in dynamic industrialization processes: Case study in the microelectronic domain*. PhD thesis, Institut Nacional Polytechnique de Grenoble.

Butdee, S. (2010). Case-based formulation to knowledge capitalization for plastic injection mold design. In *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, pages 934 – 939.

Butdee, S. (2011). Knowledge engineering and capitalization for injection mold design. In *3rd International Conference on Information and Financial Engineering*, volume 12, Singapore.

Castillo-Barrera, F. E., Medina-Ramrez, C., and Duran-Limon, H. A. (2011). Knowledge capitalization in a component-based software factory: a semantic viewpoint. In Osorio, M., Zepeda, C., Olmos, I., Carballido, J. L., Arrazola, J., and Medina, C., editors, *LA-NMR*, volume 804 of *CEUR Workshop Proceedings*, pages 105–114. CEUR-WS.org.

Chongsringam, P. and Prompoon, N. (2008). Process model design for knowledge management in cmmi organization. In Sullivan, K., editor, *5th International Conference on Intelectual Capital, Knowledge Management & Organizational Learning*. New York Institute of Technology.

Dingsyr, T. (2002). Knowledge management in medium-sized software consulting companies. *Empirical Softw. Engg.*, 7(4):383–386.

Dutoit, A. H. and Paech, B. (2000). *Handbook of Software Engineering and Knowledge Engineering*, chapter Rationale Managment in Software Engineering. World Scientific Publishing Company.

Franca, J., Baiao, F., and Santoro, F. (2012). Towards characterizing knowledge intensive processes. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 497–504.

Grundstein, M. (2000). *Knowledge Management: Classic and Contemporary Works*, chapter From Capitalizing on Company Knowledge to Knowledge Management. MIT Press.

Grundstein, M. and Rosenthal-Sabroux, C. (2005). A process modeling approach to identify and locate potential crucial knowledge: The gameth framework. Open Access publications from Universit Paris-Dauphine urn:hdl:123456789/4135, Universit Paris-Dauphine.

Grundstein, M., Rosenthal-Sabroux, C., and Pachulski, A. (2003). Reinforcing decision aid by capitalizing on company's knowledge: Future prospects. *European Journal of Operational Research*, 15:256–272.

Gueraich, S. and Boufada, Z. (2011). Building a project memory using semantic design rationale process. *Journal of Software Engineering and Application*, 4:356–370.

Iuliana, C. (2009). A knowledge management practice investigation in romanian software development organizations. *W. Trans. on Comp.*, 8(3):459–468.

Kavitha, R. K. and Ahmed, I. (2011). A knowledge management framework for agile software development teams. In *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pages 1–5.

Kokkoniemi, J. (2006). A preliminary model for generating experience knowledge based artifacts. In *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 7, pages 153b–153b.

Kokkoniemi, J. and Harjumaa, L. (2009). From software documents to experience knowledge based artifacts. In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pages 1–9.

Komi-Sirvio, S., Mantyniemi, A., and Seppanen, V. (2002). Toward a practical solution for capturing knowledge for software projects. *Software, IEEE*, 19(3):60–62.

Marcandella, E., Durand, M., Renaud, J., and Boly, V. (2009). Past projects memory: Knowledge capitalization from the early phases of innovative project. *Concurrent Engineering: Research and Applications*, 17(3):213–224.

Matta, N., Ermine, J. L., Aubertin, G., and Trivin, J. (2001). How to capitalize knowledge with the mask method? In *Workshop on Knowledge Management and Organizational Memories*.

Morello, B., Rasovska, I., and Zerhouni, N. (2005). Knowledge capitalization in system of equipment diagnosis and repair help. In *Workshop on Knowledge Management and Organizational Memories*.

Panagiotou, D. and Mentzas, G. (2011). Leveraging software reuse with knowledge management in software development. *International Journal of Software Engineering*, 21:693–723.

Rasovska, I., Chebel-Morello, B., and Zerhouni, N. (2008). A mix method of knowledge capitalization in maintenance. *Journal of Intelligent Manufacturing*, 19:347–359.

Renaud, J., Lefebvre, A., and Fonteix, C. (2004). Improvement of the design process through knowledge capitalization: an approach by know-how mapping. *Concurrent Engineering: R&A*, 12(1):25–37.

Rodriguez-Rocha, B. D., Castillo-Barrera, F. E., and Lopez-Padilla, H. (2009). Knowledge capitalization in the automotive industry using an ontology based on the iso/ts 16949 standard. In *Proceedings of the 2009 Electronics, Robotics and Automotive Mechanics Conference (cerma 2009)*, CERMA '09, pages 100–106, Washington, DC, USA. IEEE Computer Society.

Sarirete, A. and Chick, A. (2008). Knowledge capitalization in the communities of practice of engineering. In *EACP Workshop: Learning Environments for Communities of Practice*.

SEI (2010). Cmmi for development, version 1.3. Technical report, Software Engineering Institute.

Simon, G. (1996). Knowledge acquisition and modeling for corporate memory lessons learnt from experience. In *Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.

Tseng, T. and Huang, C. (2005). Capitalizing on knowledge: a novel approach to crucial-knowledge determination. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(6):919 – 931.

Vasumathy, M. (2012). The impact of knowledge management in practices in software engineering. *International Journal of Computer & Technology*, 3:500–502.