A Method to Evaluate Knowledge Resources in Agile Software Development

1st Raquel Ouriques
Department of Software Engineering
Blekinge Institute of Technology
SE 37179 Karlskrona, Sweden
raquel.ouriques@bth.se

2nd Ricardo Britto
Ericsson
SE 37133 Karlskrona, Sweden
ricardo.britto@ericsson.com

3rd Krzysztof Wnuk
Department of Software Engineering
Blekinge Institute of Technology
SE 37179 Karlskrona, Sweden
krzysztof.wnuk@bth.se

4th João Felipe Ouriques
Ericsson
SE 37133 Karlskrona, Sweden
joao.felipe.ouriques@ericsson.com

5th Tony Gorschek
Department of Software Engineering
Blekinge Institute of Technology
SE 37179 Karlskrona, Sweden
tony.gorschek@bth.se

***Abstract*—Background:** Organizations adopting Agile Software Development (ASD) use different Knowledge Management (KM) practices to retain and share knowledge. However, it is often the case that knowledge retention is carried out in an ad-hoc way.
Aims: In this study, we report our experience from proposing the Knowledge Critically Evaluation Method (KCEM) to evaluate knowledge items (KIT). Our main goal with KCEMs is to support companies to systematically retain knowledge in ASD contexts.
Method: We conducted an improvement case study to develop and evaluate KCEM. This research follows the guidelines for technology transfer between industry and academia. The case and unit of analysis is Ericsson, a Swedish company that develops telecommunication solutions.
Results: In this paper, we provide initial results of both lab and static validation, enriched by the lessons learned.
Conclusions: The preliminary results show that KCEM is easy to understand and use, provides a different perspective on the KIT by visualizing in the criticality chart, and reduces the level of abstraction associated to a knowledge subject area.

***Index Terms*—Agile Software Development, Critical Knowledge, Knowledge Management**

I. Introduction

Software development is knowledge-intensive, and knowledge is one of the key resources of many development organizations. At the same time, the software development landscape frequently changes, demanding from development organizations to learn new skills, apply new business models, and development paradigms [1]. This pressure to obtain new knowledge is then combined with the continuous degradation of the existing knowledge into obsolescence [2]. Moreover, many development organizations utilize tacit knowledge rather than explicit knowledge. Tacit knowledge is recognized as the most important resource in an knowledge-intensive activity such as the design of a new product or the code itself [3]–[6]. The domain knowledge and rationale behind software design decisions often remain tacit and undocumented.

Organizations manage their knowledge resources through codification and personalization strategies [7]. Codification relates to the activity of storing knowledge into artifacts (e.g., Wiki-based tools) when the knowledge nature tends to be explicit; personalization relates to socialization activities to communicate knowledge (e.g., team meetings and discussion groups) when the knowledge nature tends to be tacit.

Most of the Knowledge Management (KM) practices used in the Agile Software Development (ASD) context follow the personalization strategy, which rely on people socializing for sharing knowledge [8]. This strategy, while effective regarding flexibility and agility, might not scale for several teams [9]. In this scenario, knowledge can be lost or not reused by other teams. On the other hand, databases used as a codification strategy can also be filled with outdated and irrelevant knowledge [10].

Although software development organizations utilize several KM practices to retain and share knowledge, what is still missing is a way to evaluate the criticality[1] of a knowledge item [8], [11]–[14].

To address this gap, we propose **KCEM**. Its objective is to support companies to systematically retain knowledge in ASD contexts. It also contributes to avoid the accumulation of unnecessary artifacts. We designed KCEM to be lightweight and easily applicable by practitioners.

In this study, we report the initial results of a larger and longer-term research, which is to develop a scalable KM solution for companies adopting ASD. Our initial results refer to the developed method to evaluate the criticality of knowledge items in ASD contexts. Thus, we answer the following research questions in this paper:

[1] The degree to which a knowledge is essential to what it has being applied.

978-1-7281-2968-6/19/$31.00 ©2019 IEEE

- RQ1: How to evaluate knowledge criticality in agile software development contexts?
- RQ1.1: What aspects of criticality should be considered?
- RQ1.2: Which rubrics should be applied to the aspects?

The main contributions of this paper are:

- A method to support the evaluation of a knowledge item's criticality.
- Preliminary results of lab and static validations.
- The discussions presented as lessons learned.

This paper is organized as follows: In Section II, we present a brief background along with the respective related work. In Sections III,IV,V we describe respectively our research design, the proposed method, and the preliminary results of the method validation. In Section VI we discuss lessons learned on our validation. Finally, in Section VIII, we present our concluding remarks and view on future work.

II. Background and Related Work

Knowledge is a concept extensively discussed in the literature in terms of nature and understanding [6]. Since this discussion is not our main purpose, we adopt the a simple knowledge definition, based on Hislop [15], as the collection of information that provides guidance .

The two types of knowledge are: tacit, which is individual deriving from experiences, values and routines; and explicit, which is already systematized in formal language including guidelines, instructions or books [16].

Companies manage their knowledge resources through two main strategies: codification and personalization [7]. Codification relates to the activity of storing knowledge into artifacts (e.g., Wiki-based tools) when the knowledge nature tends to be explicit. It has a strong focus on the use of information technology mainly because they aim at reusability and broad access.

Personalization strategies relates to socialization activities that focus on person-to-person to communicate knowledge (e.g., team meetings and discussion groups) when the knowledge nature tends to be tacit. The strategies do not prioritize information technology since human interaction is the main objective [7].

Knowledge-based resources may have different levels of importance for different companies [6]. The decision on to retain the knowledge and how, depends on the criticality of the knowledge. There are few studies available in the literature which explore knowledge evaluation methods [17]–[19]. They are usually either extensively complex or focus mostly on generic criteria that do not fit properly specific contexts.

Tseng and Huang [18] used modeling knowledge requirements to create KM systems. The approach is limited by the context where it is used and also restricts the evaluation to explicit knowledge. However, it provided us with insights regarding the utilization of quadrants to classify the knowledge in terms of criticality.

Ermine et al. [17] utilized a generic grid to evaluate knowledge mostly applied in French companies. It is a private method, thus, not available in the literature. Even so, we still got inspired by its thematic axes to create our categories and rubrics for our method, considering the team feature, which is a particularity of most ASD methods.

Based on these studies, as well as on the output from a preliminary seminar with the company, we created the categories and rubrics for analyzing knowledge criticality and the strategy for the classification (see Section IV).

III. Research Design

We conducted an improvement case study [20] to develop and evaluate KCEM, and improve the decision regarding knowledge retention. Furthermore, this research follows the guidelines for technology transfer between industry and academia proposed by Gorschek et al. [21]. The case and unit of analysis is Ericsson[2], a Swedish company that develops telecommunication solutions.

The aforementioned model suggests a set of seven flexible steps for researching in collaboration with industry. We focus on the five first steps of the model to introduce the initial results of our research. In the first two steps, we conducted a workshop at Ericsson, which had two main purposes: i) present the results of a previous study regarding knowledge-based resources in ASD; and ii) elicit the potential issues that future research could address. As results of this workshop, we identified potential improvement areas, agreed on a research agenda, and prioritized the areas.

Data collection. To evaluate KCEM, we collect data in two phases of the technology transfer model: lab validation (**step 4**) and static validation (**step 5**). In both cases, we gather feedback from the participants regarding the list of questions for the defined rubrics, as well as the degree of agreement between the rubrics and the real world.

Data analysis. Since we collect feedback in both validation phases (detailed in V-A and in V-B), we refine KCEM accordingly for the next phase. The issues raised during the static validation are discussed in the form of lessons learned in section VI.

IV. The knowledge criticality evaluation method - KCEM

We define knowledge criticality as *the degree to which a knowledge item is essential to what it has being applied.* The method evaluates Knowledge Items (KIT), which we define as **the particular unit of knowledge**. A KIT can be different things such as a process and its associated knowledge. A KIT can be either tacit or explicit, e.g., coordination and interpersonal skills or knowledge stored by developers in wikis.

[2] www.ericsson.com

TABLE I: Rubrics for evaluating knowledge criticality

Category	Rubric	Description
Relevance	Potential to cause delay	Relate to dependency of this specific knowledge to support different tasks/activities that might get delayed if the knowledge is not available
	Obsolescence	Potential for the knowledge item become obsolete
	Adaptability	Relates to the extent that the knowledge item has potential be adjusted to different conditions
	Dependency	Dependency on the knowledge to perform the tasks of a specific or more roles
	Potential cross-benefit	Several teams would benefit from having this knowledge available
Scarcity	Available experts	Relates to the number of experts , which owns a specific knowledge, and could supply the need for it in case of, e.g., a turnover
	Originality	The knowledge is somehow original. It was not owned by the company before
	Difficulty of externalization	Relates to the easiness of representing this knowledge item in an artifact (e.g., guidelines, instructions)
	Complexity	Relates to how complex this knowledge is when comparing to the ones used in daily activities
	Staff turnover	Relates to the rate of staff turnover

To illustrate an example of KIT, let's consider the **Build Master**. This role may have many different KITs, such as *Expertise in git*[3], *proficiency on the company release process*, and *proficiency on the process of fixing software faults*.

Criticality is divided into two categories, relevance and scarcity, which we define as follows:

- *Relevance* - potential for satisfying a particular need, e.g., within a team or larger agile setups.
- *Scarcity* - relates to the extent that a KIT is readily obtainable, e.g., the number of experts that has a specific skill.

Each category has a set of rubrics (see Table I), which can be used by an individual or by a group (in a planning poker style). Each rubric is a statement/question whose answers represent a rating (e.g. agreement, likelihood, and importance) relationship, thereby we could treat them as Likert items, with values ranging from 1 to max, with $max \in \mathbb{N}$, e.g., a scale with five levels, one could answer 1, 2, 3, 4, or 5.

Since we have Likert items with different levels, the answers of each rubric r are normalized to the range $[0, 1]$ by applying a general formula $score(r) = \frac{n-1}{max(r)-1}$, where $max(r)$ is the maximum value for the scale related to the rubric r, and n is the actual answer. This normalization allows the calculation (sum of the values of all rubrics of each category) of scores for both Relevance and Scarcity, which facilitates interpretation of the measurements. Since both categories comprise five rubrics, both scores range from 0 to 5 and, in order to define the four quadrants, we use 2.5 as middle point. Note that we use radar charts to facilitate the visualization of the results of the rubrics of each category.

The result of the calculated scores is a pair $crit = (rel, sca)$ visualized in a 2-dimensional chart (see Fig. I), which has four quadrants for recommendation categories for managing the evaluated knowledge. The quadrants are described as follows:

- **Q1 - Slight criticality**. The KIT has low scores for both scarcity and utility, i.e., it has low complexity and probably will be readily available when needed.
- **Q2 - Low criticality - Vigilant**: The KIT's utility is low, but high scored in scarcity, which might be challenging to acquire it quickly. It might not have, e.g., homogeneous importance across teams or to a development process. Thus, the recommendation for this knowledge item is to make efforts to, e.g., duplicate or make it redundant on occasions where and when it is needed.
- **Q3 - Medium criticality - Handy**: The knowledge item is mostly available and important. The recommendation is to facilitate accessibility through efficient search processes, either through socialization or codification processes. For example, codify knowledge in artifacts (instructions, guidelines) or, in case of tacit knowledge, make the source visible.
- **Q4 - High criticality - Strategic**: The knowledge item is crucial and potentially contributes to continuous readiness to absorb unforeseen changes in agile contexts. Thus, it is strongly recommended to establish knowledge retention practices for this knowledge item.

[3]A broadly used software for versioning control

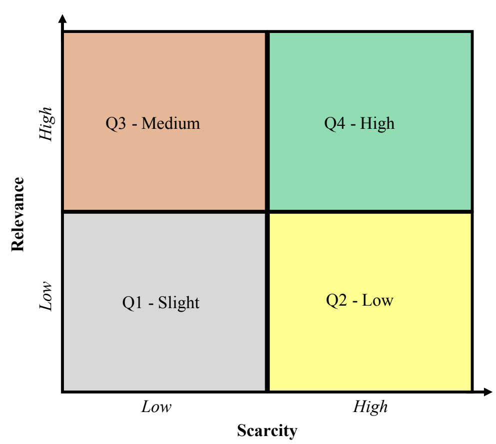

Fig. 1: Knowledge criticality

V. Validation

We performed a two-step validation. We briefly explain the lab validation, and discuss the static validation in the following sections.

A. Lab Validation

Five software engineering Ph.D candidates participated in the lab validation. They utilized the method to evaluate KITs related to their context, including knowledge about

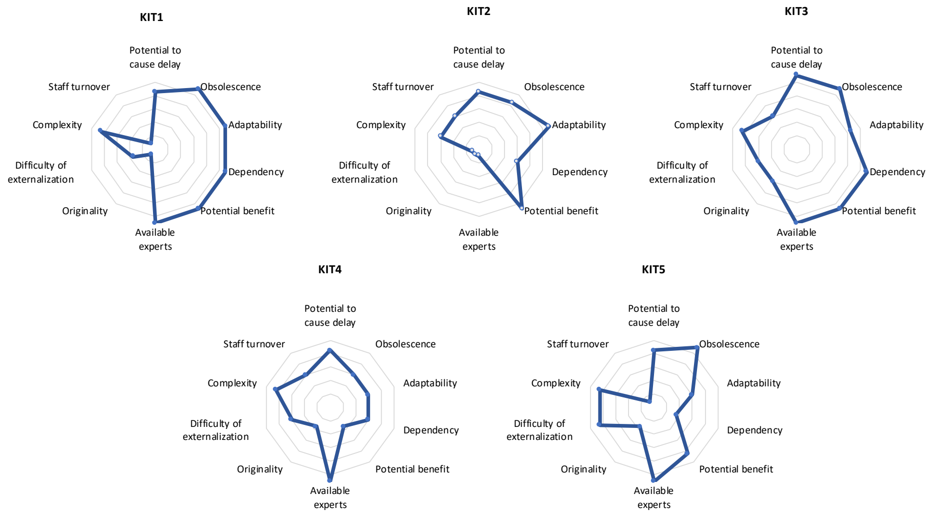

Fig. 2: Rubrics results for the KITs

how to narrow down research questions and report travel expenses.

We collected early feedback from the students regarding usability of the method and about the clarity of the questions.

B. Static Validation

In the static validation, we utilized the company´s documentation that described the development processes to collect the critical knowledge associated with each process. This step was performed together with a practitioner from the company.

Afterward, a representative of each process utilized the method and gave us feedback. In total, four line managers that are responsible for each process utilized the method. Each of them evaluated one KIT, except one that felt comfortable also evaluating the fifth one.

We selected five critical knowledge items to be evaluated by the practitioners participating in the workshop. Each KIT belongs to a different development process from one of the Ericsson's products. They are:

- KIT1: How to design test cases
- KIT2: Do trouble report handling
- KIT3: Do emergency pack
- KIT4: Start new technical product group
- KIT5: Pre-product customization development

The practitioners evaluated the KITs in approximately 10 minutes and for each of the items, we have generated radar charts (see Fig. 2) to supplement the knowledge criticality chart, see Fig. 3.

Besides the evaluation of the questionnaire, we collected feedback regarding the degree of agreement of the method to the real world. The group discussion aimed at validating the following items:

To what extent the method helps on recommending how to manage the knowledge item. The practitioners agreed to a large extent that the method is good to evaluate KITs under the KM perspective. However, as a supplement to the method they recommended that at the same time that they evaluate the criticality, they could also answer about the current status of that KIT in terms of management.

Does the method provide a somewhat different perspective about the knowledge item?. The practitioners reported a good experience using the method in two main aspects: 1 - The visualization of the method gives a different perspective about the KIT and facilitates comparing the criticality of different KIT within the same or different processes; 2 - The easiness of use was also mentioned by the practitioners, which corroborate with one of our objective of developing a lightweight method to easy application.

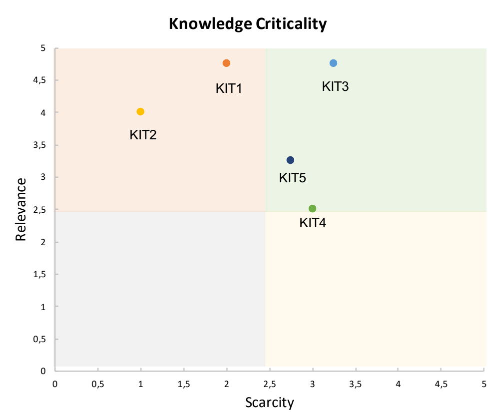

Fig. 3: Knowledge Criticality of the KITs evaluated

To what extent the results seems accurate. Even though we were not able to breakdown the processes in detail, the practitioners confirmed the accuracy of the results during the discussion. For example, while presenting the results, the two practitioners that evaluated KIT1 and KIT2 confirmed that the scarcity for those KITs are explained by the fact that the knowledge to perform those tasks is quite available for when it is needed, so its scarcity is low, and the relevance is quite high for the development processes.

VI. Lessons Learned

Overall, the method was received as quite simple to apply, with some difficulties to gather the KITs. The most important lessons learned are the following:

> Lesson #1 Gathering the KITs may be time consuming depending on how large is the analyzed process.

We utilized the Ericsson's high level document that depicts the main development processes and a few subprocesses to validate the method. However, we plan to utilize cognitive task analysis (CTA) for dynamic validation as it offers a more complete picture of the knowledge required to accomplish each task in the sub-processes. CTA is a set of methods that aim to capture and understand how cognition support people in accomplishing their tasks [22]. By applying this technique, it is possible to map tasks in companies that do not have a structured and detailed process. The second lesson learned:

> Lesson #2 The level in which the KIT is evaluated has to be defined upfront to avoid confusion regarding the context of the item.

We noticed that the development processes evaluated is quite extensive due to the size of the product, and some practitioners were confused to consider which context they should evaluate. For example, when considering the rubric "dependency", they should evaluate considering the task in relation to the deliverable, process or product in general? We were able to discuss and define the context this during the validation, however it is an important aspect to consider for researchers and other practitioners who want to apply the method.

> Lesson #3 The relevance scores tend to be biased because practitioners believe that all knowledge is highly relevant.

The practitioners confirmed that they tend to overrate their tasks and its knowledge associated. The way we phrase the questions, specially regarding relevance, can provide unreal results.

> Lesson #4 Allowing the method to have multiple entries, practitioners can evaluate the same item simultaneously.

There might be disagreements among practitioners when evaluating the KITs. However, we do not consider this a problem to our method. Since we adopt the concept that knowledge is socially constructed [23], allowing multiple entries to the same KIT will promote relevant discussion towards a better strategy to manage the KIT.

> Lesson #5 The evaluation should be followed by questions if this KIT has already strategic plan to be managed.

One of the improvements that can increase the utility of the method during the discussion of the results is to add extra questions aiming at evaluate the current practice to manage the KIT. Thus, when visualizing the results, the practitioners can also see the current status of the KM practice related to the KIT evaluated.

> Lesson #6 Having arbitrary intervals raise the need for analyzing the rubrics individually.

We chose to use arbitrary intervals to classify the KITs in method. However, we recommend that when the result falls in the border to other quadrant, practitioners should examine carefully rubric by rubric in the radar charts. For example, the KIT4 (see Fig. 3) is close to Q2. In this case, the KM practice adopted to manage the KIT can vary dramatically. The company might spend many resources to retain knowledge that it is not widely critical or the contrary, retain the knowledge locally, e.g., in one team, when it is critical to several parts of the company.

VII. Validity Threats

We follow the guidelines recommended by Runeson and Höst [20], which classify the threats in external, internal, reliability, and construct.

External validity relates to the extent that the results can be generalized. Although the results and lessons learned are limited to Ericsson's case, we believe that the rubrics are generic enough to be utilized in other software development companies adopting ASD.

Internal validity relates to the awareness about other factors that could affect the phenomenon investigated. The experiences reported could be different if more people from the same process have used the method. However, this threat does not represent a high risk because the method does not seek agreement, but allows discussion with different points of view instead, so a better KM practice is adopted in the end. Besides, we have discussed the findings with representatives of each analyzed process.

Reliability relates to how repeatable is the study. The results of the application of the method will vary from company to company, especially because they will probably identify different knowledge items as input. However, to make the research reproducible, we detailed the all the steps carried out in our investigation.

Construct validity relates to the extent that the measures utilized reflect what both researchers and the population investigated have in mind. To mitigate this threat, we explained the meaning of each rubric and category before applying the method, and also discussed the obtained results.

VIII. Conclusion

In this paper, we proposed a method for evaluating the knowledge criticality of knowledge items in ASD contexts. We have conducted a preliminary validation of the method, which was divided in two parts: lab validation and static validation. Both validations show promising results.

Considering that the main goal of our research is to support companies, the results of the static validation are of special interest. The practitioners who used our method highlighted three main advantages: easy to understand and use, provide a different perspective on a KIT by visualizing the criticality chart, and reduce the level of abstraction associated to a knowledge subject area.

The findings of this study have important implications for future KM practices including changing the way we perceived knowledge towards a resource and reducing the level of abstraction of the topic.

This research is part of an ongoing research at Ericsson. Our long term objective is to pack a scalable KM solution for companies adopting ASD. As the next steps in this research, we aim the following:

- Perform the two remaining steps of the technology transfer model. Regarding the dynamic validation step, we plan to conduct through larger pilots involving more teams. In relation to the final step, Release the solution, we plan to roll out the method at Ericsson.
- Develop an approach to help managers relate the results of using our method and existing competence in the company.
- Automate partially or entirely the criticality analysis of KITs. This might help to identify changes regarding relevance and scarcity in a more timely way.

Acknowledgment

We would like to acknowledge that this work was supported by the KKS foundation through the S.E.R.T. Research Profile project at Blekinge Institute of Technology.

We would also like to thank Ericsson and the employees involved in and being sincerely interested in our research.

References

[1] K. Wnuk, T. Gorschek, and S. Zahda, "Obsolete software requirements," *Information and Software Technology*, vol. 55, no. 6, pp. 921 – 940, 2013.

[2] A. N. K. Chen and T. M. Edgington, "Assessing value in organizational knowledge creation: Considerations for knowledge workers.," *MIS Quarterly*, vol. 29, no. 2, pp. 279 – 309, 2005.

[3] J. Barney, "Firm resources and sustained competitive advantage," *Advances in Strategic Management*, vol. 17, pp. 203–227, 2000. cited By 46.

[4] R. Glazer, "Measuring the knower: Towards a theory of knowledge equity.," *California Management Review*, vol. 40, no. 3, pp. 175 – 194, 1998.

[5] O. Steen, "Practical knowledge and its importance for software product quality," *Information and Software Technology*, vol. 49, no. 6, pp. 625–636, 2007.

[6] R. M. Grant, "Toward a knowledge-based theory of the firm," *Strategic Management Journal*, vol. 17, no. S2, pp. 109–122, 1996.

[7] M. Hansen, N. Nohria, and T. Tierney, "What's your strategy for managing knowledge?," *Harvard Business Review*, March-April 1999.

[8] R. Ouriques, K. Wnuk, R. B. Svensson, and T. Gorschek, "Knowledge management strategies and processes in agile software development: A systematic literature review," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 3, pp. 345–380, 2019.

[9] T. Dingsøyr and D. Smite, "Managing knowledge in global software development projects," *IT Professional*, vol. 16, no. 1, pp. 22–29, 2013.

[10] L. F. Lai, "A knowledge engineering approach to knowledge management," *Information Sciences*, vol. 177, no. 19, pp. 4072 – 4094, 2007.

[11] S. Dorairaj, J. Noble, and P. Malik, "Knowledge management in distributed agile software development," in *2012 Agile Conference*, pp. 64–73, Aug 2012.

[12] T. Chau and F. Maurer, "Tool support for inter-team learning in agile software organizations," in *Advances in Learning Software Organizations* (G. Melnik and H. Holz, eds.), (Berlin, Heidelberg), pp. 98–109, Springer Berlin Heidelberg, 2004.

[13] J. T. Karlsen, L. Hagman, and T. Pedersen, "Intra-project transfer of knowledge in information systems development firms," *Journal of Systems and Information Technology*, vol. 13, no. 1, pp. 66–80, 2011.

[14] K. Kuusinen, P. Gregory, H. Sharp, L. Barroca, K. Taylor, and L. Wood, "Knowledge sharing in a large agile organisation: A survey study," in *International Conference on Agile Software Development*, pp. 135–150, Springer, 2017.

[15] D. Hislop, R. Bosua, and R. Helms, *Knowledge management in organizations: A critical introduction.* Oxford University Press, 2018.

[16] I. Nonaka, R. Toyama, and N. Konno, "Seci, ba and leadership: a unified model of dynamic knowledge creation," *Long Range Planning*, vol. 33, no. 1, pp. 5 – 34, 2000.

[17] J.-L. Ermine, I. Boughzala, and T. Tounkara, "Critical knowledge map as a decision tool for knowledge transfer actions," *Electronic Journal of Knowledge Management*, vol. 4, no. 2, pp. 129–140, 2006.

[18] Tzu-Liang Tseng and Chun-Che Huang, "Capitalizing on knowledge: a novel approach to crucial-knowledge determination," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, pp. 919–931, Nov 2005.

[19] T. Tounkara and J.-L. Ermine, "Méthodes de cartographie pour l'alignement stratégique de la gestion des connaissances," *Hermes sciences*, p. 100, 2008.

[20] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, p. 131, Dec 2008.

[21] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for technology transfer in practice," *IEEE Software*, vol. 23, pp. 88–95, Nov 2006.

[22] B. Crandall, G. Klein, G. A. Klein, and R. R. Hoffman, *Working minds: A practitioner's guide to cognitive task analysis.* Mit Press, 2006.

[23] I. Nonaka, H. Takeuchi, and K. Umemoto, "A theory of organizational knowledge creation," *International Journal of Technology Management*, vol. 11, no. 7-8, pp. 833–845, 1996.