Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 181 (2021) 386–393

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2020

Agility and system documentation in large-scale enterprise system projects: a knowledge management perspective

Makoto Nakayama[a], Eli Hustad[b,*], Norma Sutcliffe[a]

[a]*DePaul University, Chicago, Illinois, USA*
[b]*University of Agder, 4604 Kristiansand, Norway*

Abstract

The growth of the agile approach usage comes with a deemphasis on formal documentation (explicit knowledge) and an increased reliance on personal interactions (tacit knowledge) for knowledge transfer. However, the sharing of tacit knowledge poses challenges. The agile approach is prone to knowledge hoarding, as well as knowledge loss from employee turnover and reassignment during periods of significant organizational changes. This study proposes a model that frames documentation and personal interactions as co-agents of system knowledge transfer. We report the preliminary confirmation of crucial antecedents along the dimensions of codification and personalization strategies to support our model. We present a set of findings on current practices, as well as a set of knowledge-sharing issues on system documentation based on three main categories. The first category incorporates *system development approaches* applied in large-scale enterprise systems projects. The second and the third categories comprise eight knowledge management themes, classified into the dimensions of *personalization* and *codification* for knowledge sharing and document practices. Finally, we put forward five propositions based on our findings.

Keywords: Agile system implementation; system documentation; system knowledge; tacit knowledge; explicit knowledge; knowledge sharing

* Contact Author. Tel.: +47-38-14-16-21.
E-mail address: eli.hustad@uia.no

1877-0509
10.1016/j.procs.2021.01.181

1. Introduction

The main objectives of this study are to explore current documentation practices for large-scale system development projects and to understand the usefulness and value of these documents for both consultant and client organizations (the impact of explicit knowledge), while assessing the extent of the agile approach use at the same time. Additionally, we examine how knowledge sharing and communication in system development projects can generate value and positive project outcomes (the impact of tacit knowledge). More specifically, the following research question has guided our research: *What practices of system documentation and knowledge sharing assure the retention of requisite organizational system knowledge without losing agility?* We explore the socio-technical circumstances that unfold in system documentation practices and knowledge sharing through 16 expert interviews with system consultants and project managers.

The motivation of this research derives from implementing agility in large scale projects to make them more malleable. The emphasis on agility for software development accompanies a low level of protocols, as well as a minimal level of formal documentation [1, 2]. The agile approach is found helpful for the implementation of large-scale enterprise systems (ES), such as enterprise resource planning (ERP) systems, although how to apply the agile approach remains a challenge [3]. Agile practices heavily rely on co-location and face-to-face communication. If we categorize knowledge in two ways, one is tacit knowledge rooted in the "analog" process of continuous actions and informal communication, and the other is explicit knowledge captured in "digital" records of documentation and databases [4]. The deemphasis on comprehensive, formal documentation is based on the assumption that tacit knowledge is more valuable than explicit knowledge and that the effort spent on documentation development is counterproductive [5].

However, tacit knowledge is susceptible to knowledge loss and prone to suffer from knowledge hoarding [6]. For instance, previous literature claims that ninety percent of knowledge in any organization resides in people's heads [7] as opposed to written documents. Knowledge sharing is often limited to the scope of teams [8], and employee turnover disrupts its sharing [9]. Displaced ERP staff, thus, may disrupt the current and future ERP operations significantly. Moreover, the principles of documentation for small-scale system projects (e.g., working software over comprehensive documentation) do not necessarily extrapolate those for large-scale systems [10]. Documentation is an important factor for meeting the expectations of end-users and developers in the traditional structured approach [11]. Even when using the agile approach, an adequate level of documentation is still vital at the early stages of the project, especially for requirements [12, 13]. This study contributes to understand how stakeholders of complex ES system implementation trajectories can tackle and balance documentation practices with informal knowledge sharing to keep up with agility. This is crucial for developing system knowledge that is valuable for the organization during the system's life cycle.

The paper is structured as follows. First, we present the research background in terms of system documentation, system knowledge which is summarized into a research model. Second, our research method is described, followed by a presentation and classification of our key results. Finally, we make some concluding remarks including implications and suggestions for future research.

2. Research background

2.1. System documentation

System documentation is defined as "the descriptive information that explains the system, provides the audit trail of modifications to the system, and serves as a training aid" [14], where nine criteria determine the quality of documentation: readability, completeness, accuracy, ease of update, change tracking, functionality, identification of responsibility, identification of authority, and organizational documentation standards.

The "waterfall" approach, which is typically applied in larger companies, produces system documents, such as software requirements, design documents, test plans, and operating instructions (users' manual) [15]. The system documentation provides an explicit means of communication that clarifies specifications, designs, and logical descriptions of how systems work [16]. These documents minimize the risks associated with contractual vendor service delivery [17], support development, maintenance, management of decisions [18], and compliance fulfillment

[19]. The challenges of documentation are the costs of production, maintenance, and update, as well as the time and effort spent on searching and reading documents [18].

Large-scale information system (IS) projects, such as ERP projects, have a strong tradition of applying a rigid and stage-gate implementation approach. Especially in large-scale projects, mistakes at one or more stages of the system development lifecycle can be harmful to companies. Even if the stage-gate model allows returning to an earlier stage, going back to a previous stage involves costly rewrites for both documentation [20] and development of the application (e.g., configuration or customization of ERP systems). Although ERP systems are standardized, off-the-shelf software packages that build on best practices based on different industries, the adoption of ERP systems is a complex organizational and technical undertaking. The ERP system delves deeply into the business, aiming to control all organizational resources and transactions through one system. Technically, the systems are challenging to implement in terms of configuration, adaptation, and data conversion from legacy systems. Organizationally, for optimal use, the systems introduce changes in business processes, work routines, and roles of the employees [21]. ERP projects are risky endeavors and have a reputation for failures and exceeding implementation budgets [22]; thus, appropriate documentation practices are crucial, especially in the beginning of the project, to secure preparedness and avoid problems later on in the ERP lifecycle [23].

2.2. Managing system knowledge

We use a knowledge management (KM) perspective by combining the *personalization* (behavioral, networking) and the *codification* (technocratic, repository) strategies for knowledge work [24-26]. We define system knowledge as the relevant awareness, understanding, and conceptual capabilities for planning, designing, developing, implementing, maintaining, and utilizing IS in the organization. We view documentation forms of system knowledge as explicit (codified) knowledge, whereas non-documented forms comprise tacit knowledge. While explicit knowledge can be shared through documentation, the transfer of tacit knowledge involves a complex sequence of interactions between individuals [27]. However, not all system knowledge is codifiable or documented due to production and maintenance costs. Regardless of knowledge forms, knowledge transfer relies on individual motivations, formal and informal networks, and social cohesion [28].

The personalization model (tacit knowledge) seeks to support knowledge-sharing and rich communication activities in different types of groups and communities (e.g., system development teams, the ERP project organization) by providing them with interactive communication tools, such as email, videoconference, intranets, and social software. The personalization model focuses on tacit knowledge and informal relations in organizations[29]. In contrast, the codification model is more technologically centered and pays attention to how to codify knowledge by creating documents for reuse purposes through repositories [25].

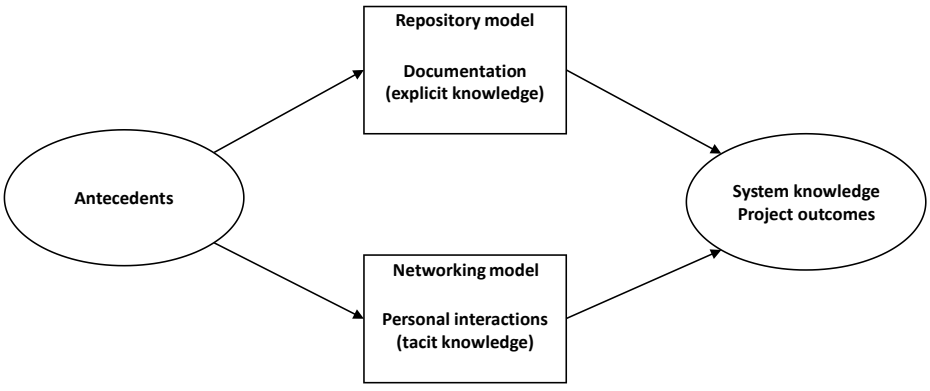

Fig. 1. Conceptual research model

Knowledge transfer also depends on power and having strong or even weak ties relations, which require mutual understanding, negotiated roles, and knowledge sharing [30]. Another factor is networking behavior, which is influenced by team type, composition, and culture [31]. Thus, we conceptualize our preliminary high-level research model in Figure 1. We argue that this model is appropriate for understanding how both tacit and explicit knowledge influence the entire system development lifecycle. Our aim is to explore how both personal interactions (tacit knowledge-networking model) and documentation practices (explicit knowledge-repository model) contribute to a successful IS outcome, which has embedded knowledge value for both client and consultant organizations.

3. Research method

While we have a general research model, how documentation use and personal interactions impact system knowledge and project success often depends on the types of IS and projects. For this study, we chose to focus on relatively large-scale enterprise IS that support the core operations of firms. We used the expert interview approach [32] as an exploratory tool to identify specific variables and their relationships for theory-generating purposes. We interviewed eight experienced consultants since system consultants typically possess knowledge of multiple organizations and their systems. Additionally, we interviewed eight seasoned industry experts who possess knowledge within large organizations. Most of the experts had experience in large-scale ERP projects. Table 1 lists the industries and the roles of our interviewees. The project methodologies used mostly involved the agile approach for smaller systems and the development phase, whereas more structured approaches were observed for the planning and analysis phase and the implementation of larger systems, such as ERP systems. Each interview was recorded, transcribed, and coded.

Table 1. Industry and role profile of each expert.

#	Industry	Role	#	Industry	Role
1	Agrochemical, services	Consultant	9	Food processing, services	Project director
2	Higher education, materials	Consultant	10	Pharmaceutical	Project director
3	Various industries	Consultant	11	Various industries	Consultant
4	Financial services	Developer	12	Electronics and office equipment	Project director
5	Various (*CRM systems)	Consultant	13	Various industries (ERP)	Consultant
6	Financial services	Project participant	14	Engineering and construction	Internal ERP consultant
7	Various (**ERP systems)	Consultant	15	Engineering and construction	Internal ERP consultant
8	Various (ERP systems)	Consultant	16	Power and energy	Internal ERP consultant

*CRM: Customer relationship management systems **ERP: Enterprise resource planning systems

4. Results

In this section, we describe the key findings of this study. Table 2 lists eleven themes classified into three main categories. The first category encompasses different *system development approaches* applied in large-scale IS projects, which were highlighted by the experts during the interviews. The second and the third categories comprise eight KM themes, classified into the dimensions of *personalization* and *codification* for knowledge sharing and document practices, respectively. Each theme is briefly described in the following subsections, along with supporting evidence from the interviews.

4.1. System development approaches and documentation

While large-scale IS projects usually have a tradition of applying waterfall approaches with rigorous documentation procedures, we identified a trend of an increasing use of more elements from agile methods, even in ERP projects. In general, high-level documentation, as well as less text with more figures, diagrams, and videos, were preferred.

Collaboration and trust are also required for knowledge to be shared across organizations [33]. For example, it is important to trust in the consultant-client relationship to enhance knowledge transfer in complex ERP system projects. A study shows that knowledge transfer between consultants and clients relies on communication-related, motivational, and knowledge-related factors [34].However, several interviewees reported challenges with transferring to an agile approach because demanding changes in work procedures for the whole IS project team occurred when adapting to a new system development methodology. Our findings revealed that a mixed approach of both waterfall and agile methods seemed the most appropriate for large- scale IS projects. Documentation was important, especially at the beginning of the project, when developing requirement specifications, as well as for formalizing decision points during the project lifecycle. The documentation practices were also related to the organizational context, and compliance with certain standards was the case for specific domains. However, the importance of a workshop that supported knowledge sharing, discussions, and lessons learned was highlighted among the interviewees utilizing a mixed approach. In particular, postmortem sessions after each stage of the project and at the final stage were highlighted as crucial for building new knowledge and learning outcomes of the team collaboration. Postmortems were also important for transferring experiences to the next project. In the following subsections, we present our key findings to highlight the practices from both agile and traditional approaches with respect to documentation.

Table 2. Themes classified into three main categories.

Themes identified in the data
System development approaches and documentation
Stage-gate and waterfall – rigid documentation practices
Agile approaches – communication, knowledge sharing, informal documentation
Hybrid approaches – combining waterfall and agile – high-level documentation and communication events, postmortem sessions
**KM perspective – Personalization (tacit) dimension*
Organizational size and culture
Face-to-face interactions
Knowledge hoarding
KM perspective – Codification (explicit) dimension
Legacy documentation
Compliance and contractual obligation
Quality and variety of documentation
Details of documentation
Embedded documentation

*KM: knowledge management

4.2. Personalization (tacit) dimension

Organizational size and culture. The variety of documentation practices is often rooted in organizational nature. One interviewee, who belonged to a unit with an informal culture within the formal organizational culture, stated, "I'm definitely at a documentation-light shop" (4). (The reference numbers in parentheses correspond to those in Table 1.) Organizational size and complexity were factors that affected ERP implementation: "smaller companies … were promised to implement [that the ERP system would be implemented] in four months. They don't rely on documents" (7). Larger companies "required a lot more structure and a lot more documentation, just because of the sheer size of a full ERP implementation … in a few smaller internal projects to just develop … more a scrum, flexible, and much, much, much less documentation" (2).

Face-to-face interactions. One important finding is that documentation should be complemented with personal interactions. "I tried to have face-to-face interaction and sit down with people, especially developers" (6). "If you wanted to bring somebody else up to speed on something, you might sit and tell them some pieces about it ... [the]

more I learned by … having somebody coach through, not too much relying on the documentation" (9). However, documentation is a valuable means to overcome the limitations of asynchronous communication across different languages and time zones. "Because we're in a very global environment, our colleagues really depend on documentation in order to stay updated on projects …" (10). "When we customize an ERP solution, we apply more agile elements, and the dialogues with the customers are the most important … we apply lessons learned sessions and postmortems and share internally [on a global scale] with the whole ERP group" (13).

Knowledge hoarding. Many of the interviewees were using a SharePoint-based central documentation repository. However, the inadequate legacy system documentation necessarily leads to several issues: "You'll find a lot of times it's also in the people's head[s]" (1). Moreover, intentional, or unintentional knowledge hoarding was noted. "[Y]ou typically have several subject matter experts who are the bottleneck … it's just a lot of just knowledge within the heads of the people" (5). "[I]f a person was to leave, all that knowledge [would leave] with him" (6). "My predecessor left three months before I started, and there was no transfer of skills from him and very little that was documented … had problems with figuring out what [was] going on … what [was the] custom" (14). "You get very dependent [on] clever consultants … it is not easy for the one working on support to figure out things that happened in the past, with no documentation" (15).

4.3. Codification (explicit) dimension

Legacy documentation. Regarding documentation on current and past systems, the vast majority of the interviewees noted that it was frequently either outdated or non-existent. "A lot of times, you'll find the current process is not documented at all" (1). "Documentation just never kept up with [enhancement and maintenance]" (4). "[T]he documentation on the old system is far less important than the documentation on the new system" (8). "[T]he lifespan of document usefulness depends on whether the component in question is volatile or stable" [11].

Compliance and contractual obligation. The common reasons for documentation creation and maintenance are legal and contractual mandates across different industries; the pharmaceutical, finance, and health sectors have high compliance obligations.

"It also can be related to compliance issues, which are also typically more geared towards more traditional larger businesses in financial services or healthcare or manufacturing, where they've got the discipline to maintain processes and practices and documentation around it. It's probably also a function of the team and how organized the team is that you're working with" (5).

"[E]very phase has a certain set of documents we produced as part of the contract" (7). "I'd probably say a 'C' [for the average company's documentation]" (3). However, "[thorough documentation] is in response to the regulatory requirements" (10).

Quality and variety of documentation. Even when documentation existed, its practices varied considerably across organizations. The value of existing documents was also debatable, and they were, in many cases, outdated. "[In my entire career], I have come across a handful of Word documents written by business analysts or product managers, but I rarely found them useful […]. Things move very quickly, and because of that, it's very hard to keep documentation up to date, and at some level, I can see why no documentation is better than out-of-date documentation" (4).

"I find [the practice of documentation] all over the place" (2). "The highly regulated companies like pharma or financial companies ... tend to have pretty good documentation" (3). "[T]he extent and quality of the documentation varies [*sic*] pretty dramatically, and we should have templates, kind of best practices templates" (5). "The documentation has most value for the person supporting the solution after go-live" (13).

Details of documentation. The cost of creating documents was frequently mentioned, and in such cases, a document was treated as a by-product. "[A] lot of times, people just don't have time" (1). "[W]e're finding that not everybody writes well; not everybody knows how to organize their thoughts on paper" (3). "I find a detailed spec doc to be overkill" (4). "Sometimes, they do new queries without being explicit about it; we should demand [from] people [externals] who implement with us to document what they are doing" (16). "… we [have] never tried to charge our customers based on what we are documenting …. Documentation is just a by-product of the services we give; it just comes out of [the process] and is [just for getting] a sign-off. As part of best practice, you should get a sign-off on

each phase, so you can go on to [the] next. That is the basic idea. There should be something you can use for training or audit and confirming ... [the] expectations you have …" (12).

Embedded documentation. A set of coding remarks can be considered an embedded form of documentation. An interesting finding involved two opposing views. On the one hand, "the codebase was well-documented and commented within the code" (6). On the other hand, "our rules for documentation and comments within the code said there must never be a need for such a thing ... if comments become necessary…, then the code itself is poorly written or too complex" (11).

5. Conclusion and implications

We have conducted an exploratory study on documentation practices and knowledge sharing in large-scale ES projects. Our study's findings are derived from 16 interviews with experts from consulting organizations and firms representing a variety of industries, and most of the experts have ERP project experiences with stage-gate, agile, and hybrid system development approaches. This study recommends a research model that frames documentation (explicit knowledge) and personal interactions (tacit knowledge) as co-agents of system knowledge transfer. Our findings comprise a preliminary confirmation of crucial antecedents along the dimensions of codification and personalization strategies to support our model. Our research contributes to the literature on documentation practices in large-scale ES projects; the system development approach seems to affect the outcome of a project of this kind. Our study also enriches the KM literature by shedding light on system knowledge in large-scale ES projects through the use of a networking/repository perspective from KM. Based on our findings, we present a general observation, followed by four propositions for the documentation practices of large-scale system implementation. Generally, ERP documentation practices under the agile approach focus on the *personalization* perspective, while those under the waterfall methodology emphasize the organizational *codification* approach.

The results imply that documentation practices are best assessed together with non-written forms of communication practices involving personal interactions, such as face-to-face meetings and workshops. Thus, our research model (Figure 1) has been confirmed at a high level.

We have identified key challenges in finding the right balance between the personalization and the codification approaches, which relate to the agile and the stage-gate methods, respectively.

Our findings indicate that *a hybrid approach* (combining agile elements with a stage-gate approach) can be an appropriate solution for large-scale ES projects (e.g., ERP projects) to create suitable system knowledge (tacit and explicit) over time. While the agile approach is applied more than ever before, especially for smaller systems and projects, no experts we interviewed denied the value of documentation at some level. What matters is under what circumstances documents are created and then used. For instance, high-level and concise documents are easier to produce and can facilitate the success of large-scale projects without losing agility. ERP projects and the use of agile approaches have constituted a contentious case, particularly in consultancy. We propose high-level abstractions of ERP specifications, the creation of internal documentation, the storage of documentation, and the importance of an organizational knowledge-sharing and communication culture.

However, research contributions and empirical evidence on ERP and agile approaches are limited [35]. Further work is required to establish the viability of agile approaches and documentation practices in large-scale ES projects. There is a need to further understand specific conditions and contextual features leading to system knowledge when organizations implement large-scale ES. For developing a full picture, future studies should refine our research model (Figure 1) to identify specific antecedents and confirm the existing antecedents found in this study. Moreover, there are still many unanswered questions about documentation practices and personal interactions in the context of organizational KM, as well as how to maintain significant system knowledge and achieve continuing project success. There is a need for further understanding of specific conditions and contextual features that enhance system knowledge when organizations implement large-scale ES, which future research should take into consideration.

References

[1] Cho, J. (2008) "Issues and Challenges of agile software development with SCRUM." *Issues in Information Systems* **9**: 188-195.

[2] Ji, F. and T. Sedano (2011) Comparing extreme programming and Waterfall project results. *24th IEEE-CS Conference on Software Engineering Education and Training* (CSEE&T), IEEE**:** 482-486.

[3] Baig, J. J. A., A. Shah and F. Sajjad (2017). Evaluation of agile methods for quality assurance and quality control in ERP implementation. *Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, IEEE.

[4] Nonaka, I. (1994) "A Dynamic Theory of Organizational Knowledge Creation." *Organization Science* **5**(**1**): 14-37.

[5] Turk, D., F. Robert and B. Rumpe (2005) "Assumptions underlying agile software-development processes." *Journal of Database Management (JDM)* **16**: 62-87.

[6] Ersoy, I. B. and A. M. Mahdy (2015) "Agile knowledge sharing." *International Journal of Software Engineering (IJSE)* **6**: 1-15.

[7] Smith, E. A. (2001) "The role of tacit and explicit knowledge in the workplace." *Journal of knowledge Management* **5**(**4**): 311-321.

[8] Kasauli, R., G. Liebel, E. Knauss, S. Gopakumar and B. Kanagwa (2017) Requirements Engineering Challenges in Large-Scale Agile System Development. *IEEE 25th International Requirements Engineering Conference* (RE)**:** 352-361.

[9] Droege, S. B. and J. M. Hoobler (2003) "Employee Turnover And Tacit Knowledge Diffusion: A Network Perspective." *Journal of Managerial Issues* **15**: 50-64.

[10] Rolland, K., T. Dingsoyr, B. Fitzgerald and K.-J. Stol (2016) Problematizing agile in the large: alternative assumptions for large-scale agile development. *39th International Conference on Information Systems*, Association for Information Systems (AIS).

[11] Nasution, M. F. F. and H. R. Weistroffer (2009) Documentation in Systems Development: A Significant Criterion for Project Success. *42nd Hawaii International Conference on System Sciences***:** 1-9.

[12] Behutiye, W., P. Karhapää, D. Costal, M. Oivo and X. Franch (2017) Non-functional requirements documentation in agile software development: challenges and solution proposal. *International conference on product-focused software process improvement*, Springer**:** 515-522.

[13] Shafiee, S., L. Hvam, A. Haug, M. Dam and K. Kristjansdottir (2017) "The documentation of product configuration systems: A framework and an IT solution." *Advanced Engineering Informatics* **32**: 163-175.

[14] Halloran, D., S. Manchester, J. Moriarty, R. Riley, J. Rohrman and T. Skramstad (1978) "Systems development quality control." *MIS Quarterly* **2**: 1-13.

[15] Royce, W. W. (1970) *Managing the development of large software systems*. IEEE WESCON. Los Angeles**:** 328-388.

[16] Osborne, W. M. (1988) "All about software maintenance: 50 questions and answers." *Information System Management* **5**: 36-43.

[17] Gonzalez, R., J. Gasco and J. Llopis (2005) "Information systems outsourcing risks: a study of large firms." *Industrial management & Data systems* **105**: 45-62.

[18] Zhi, J., V. Garousi-Yusifoğlu, B. Sun, G. Garousi, S. Shahnewaz and G. Ruhe (2015) "Cost, benefits and quality of software development documentation: A systematic mapping." *Journal of Systems and Software* **99**: 175-198.

[19] Masli, A., V. J. Richardson, M. W. Watson and R. W. Zmud (2016) "Senior Executives' IT Management Responsibilities: Serious IT-Related Deficiencies and CEO/CFO Turnover." *MIS Quarterly* **40**: 687-708.

[20] Kramer, M. (2018) "Best practices in systems development lifecycle: An analyses based on the waterfall model." *Review of Business & Finance Studies* **9**(**1**): 77-84.

[21] Strong, D. and O. Volkoff (2010) "Understanding Organization-Enterprise System Fit: A Path to Theorizing the Information Technology Artifact." *MIS Quarterly* **34**(**4**): 731-756.

[22] Hustad, E. and D. H. Olsen (2014) ERP Implementation in an SME: a Failure Case. In J. Devos, H.V. Landeghem, and D. Deschoolmeester. (Eds.). *Information Systems for Small and Medium-sized Enterprises:State of the Art of IS Research in SMEs.*, Springer**:** 213-228.

[23] Sammon, D. and F. Adam (2010) "Project preparedness and the emergence of implementation problems in ERP projects." *Information & Management* **47**(**1**): 1-8.

[24] Hansen, M. T., N. Nohria and T. Tierney (1999) "What's your strategy for managing knowledge? ." *Harvard Business Review* **77**(**2**): 106-116.

[25] Alavi, M. (2000) Managing organizational knowledge. *Framing the Domains of IT Management. Projecting the Future…Through the Past*. R. W. Zmud. Cincinnati, OH, Pinnaflex Educational Resources**:** 15-28.

[26] Boden, A., G. Avram, L. Bannon and V. Wulf (2012) "Knowledge sharing practices and the impact of cultural factors: reflections on two case studies of offshoring in SME." *Journal of software: Evolution and Process* **24**(**2**): 139-152.

[27] Li, M., H. Liu and J. Zhou (2018) "G-SECI model-based knowledge creation for CoPS innovation: the role of grey knowledge." *Journal of Knowledge Management* **22**: 887-911.

[28] McLure Wasko, M. and S. Faraj (2005) "Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice." *MIS Quarterly* **29**(**1**): 35-57.

[29] Wenger, E., R. McDermott and W. M. Snyder (2002) *Cultivating communities of practice*. Boston, Harvard Business School Press.

[30] Pozzebon, M. and A. Pinsonneault (2012) "The dynamics of client–consultant relationships: exploring the interplay of power and knowledge." *Journal of Information Technology* **27**(**1**): 35-56.

[31] Šmite, D., N. B. Moe, A. Šāblis and C. Wohlin (2017) "Software teams and their knowledge networks in large-scale software development." *Information and Software Technology* **86**: 71-86.

[32] Bogner, A. and W. Menz (2009) The theory-generating expert interview: epistemological interest, forms of knowledge, interaction. In *Interviewing Experts*. Springer**:** 43-80.

[33] Chen, Y.-H., T.-P. Lin and D. C. Yen (2014) "How to Facilitate Inter-organizational Knowledge Sharing: The Impact of Trust." *Information & Management* **51**: 568-578.

[34] Ko, D.-G., L. J. Kirsch and W. R. King (2005) "Antecedents of knowledge transfer from consultants to clients in enterprise system implementations." *MIS Quarterly* **29**: 59-85.

[35] Isetta, S. and M. Sampietro (2018) "Agile in ERP Projects." *PM World Journal* **VII - September**(**IX - September 2018**): 1-18.