2011 Sixth IEEE International Conference on Global Software Engineering Workshops

Architectural Knowledge Management Practices in Agile Global Software Development

Viktor Clerc
VU University Amsterdam
viktor@cs.vu.nl

Patricia Lago
VU University Amsterdam
patricia@cs.vu.nl

Hans van Vliet
VU University Amsterdam
hans@cs.vu.nl

***Abstract*—The management of knowledge, in particular knowledge about the architecture, plays a pivotal role in global software development (GSD). In earlier research, we have defined what architectural knowledge is, and elaborated several practices for sound architectural knowledge management in a distributed setting. In this paper, we report on a large survey performed at an agile global software development organization in which we interviewed 38 employees spread across three development sites. Our aim was to validate our earlier research by determining what practices for architectural knowledge management in GSD are used in practice. The results show that the case study organization emphasizes architectural knowledge management practices that promote decentralization and, as a consequence, personalization (as opposed to centralization viz. codification). We identified one new useful practice: "peered sites", covering a combination of activities that support a balance in decision-making power across sites.**

I. Introduction

Over the past years, we observed an increase in attention to the management of knowledge in global software development (GSD) organizations. For these development organizations, timely availability of accurate knowledge is important for effective and efficient software processes (see e.g. [1], [2], and the MARK and KNOWING workshop series, held in conjunction with the *IEEE Requirements Engineering Conference* and *IEEE International Conference on Global Software Engineering* series respectively).

A specific, yet extremely important kind of knowledge that needs to be shared is knowledge about the software architecture. In a software architecture, the global structure of the system to be built is decided upon. This structure, among others, should capture the major architectural decisions that led to it [3]. Capturing these architectural decisions facilitates a better decision-making process in shorter time, saving rework and improving the quality of the architecture [4], [5]. Hence, it is important not only to capture the resulting architectural design, but also the decisions, including their rationale, that led to that design: *architectural knowledge*.

In the GRIFFIN project [6], we have performed research in the area of architectural knowledge. We define *architectural knowledge* as the integrated representation of the software architecture of a software-intensive system (or a family of systems), the architectural design decisions, and the external context/environment. We have formalized the concept of architectural knowledge in a *core model of architectural knowledge* [7]. Possible reasons for the use of architectural knowledge have been identified and validated as well [8], [9]. In the context of GSD, sound management of architectural knowledge can help overcome the challenges innate to GSD. Architectural knowledge management can be implemented by performing a series of well-defined *practices*. In the GRIFFIN project, we have developed and initially validated these practices, see e.g. [10]–[12].

In the research described in this paper, we have validated the architectural knowledge management practices for their actual use by conducting a survey at a large agile global software development organization. After having interviewed 38 employees across three development sites and having analyzed the results, we conclude that the architectural knowledge management practices that promote decentralization get much more attention than those promoting centralization. We furthermore identified one additional useful practice, "peered sites", covering activities that support a balance in decision-making power across sites.

This paper is structured as follows. Section II provides the research question, an overview of the organization where we conducted our survey and details the approach used for this research. Next, Section III lists the results of the analysis. In Section IV, we discuss the possible limitations that apply to this research. We conclude this paper in Section V with the conclusions.

II. Research Question and Approach

In this section, we first outline our research question (Section II-A). Next, Section II-B provides an overview of the organization where we conducted the survey. In Section II-C, we describe the approach used for conducting the survey.

A. Research Question

Our primary interest lies in validating the set of practices for architectural knowledge management in global software development. In earlier research, we have developed several practices and validated their usefulness in an organization

978-0-7695-4558-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ICGSE-W.2011.17

involved in GSD [10]–[12]. Table I provides an overview of the practices, including references to the earlier research. These practices include frequent interaction across sites, prioritizing a set of rules with which the architecture and underlying decisions should comply, writing down deviations from these architectural rules in case of non-compliance (including a rationale for non-compliance), the urgent request mechanism as a light-weight, non-intrusive manner to quickly gain information on a specific topic of interest, establishing a project structure in which is it clear what the communication responsibilities are, and establishing a single configuration management system in which work products such as documentation, scripts, test cases, and source code are stored. For an elaborate desccription of each individual practice, we refer to the work referenced by Table I. The work presented here aims at a broader evaluation of the use of these practices at another large GSD organization. In addition, we are open to identifying practices that are in use at Océ Technologies but not identified through our earlier research.

Hence, our research question is as follows:

What practices for managing architectural knowledge are used?

B. Characterization of the Organization

We performed the survey at Océ Technologies. Océ Technologies produces high-end printers for the business markets in high-volume printing, wide-format printing and office printing. Océ Technologies is based in The Netherlands and has several additional development sites spread across the globe.

The software in these printers manages user interaction, renders images, and controls the print engine. Software for printers is typically developed by several software development teams, plus architects, integrators, and testers. These software teams reside at the various development sites of the organization.

A typical team structure of project teams at Océ is depicted in Figure 1. A project manager heads the project and is responsible for its planning, realization, and successful completion. The project manager also agrees upon the high-level specifications of the project with upper management and marketing personnel. Requirements and specifications are compiled into product properties by the lead architect in the team. The specifications written by the lead architect start from a user-centric view, i.e., scenarios on how the end-users will interact with the product. For instance, the architect is responsible to decide what happens in case of an interrupt request from the user of the printer (as in how should it function, which software units should be triggered and how should they function), define the interfaces between these units and the like. Teams also comprise a system integrator who integrates the different software units to build the software system. Additionally, the integrator reports issues encountered during integration and assigns them to the appropriate team or person to be addressed. All three – the project manager, the lead architect, and the integrator – are assigned to a project for its entire duration until the product has been released and are located at the Dutch development site.

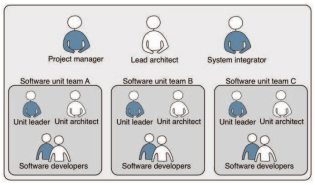

Figure 1. Typical project team structure at Océ.

Project teams also include one or more software unit teams that implement the software units. Each unit has a unit leader and a unit architect analogous to the project manager and lead architect at the project level. The unit leader is responsible for planning and organizing. The unit architect transforms the high-level specifications received from the lead architect into detailed technical specifications and pass them to the software developers who implement the code and test it. The unit architects are coordinated by the lead architect, who is often the only team member with the overall view of where (or in which units) the different functionalities of the product reside. Most software units are not developed for a single product but their deliverable is tuned and integrated into several products. A software team can develop a software unit for four, or even more, projects at the same time. This challenges system behavior as well as architecture.

As a basis for our research, we selected two large projects. These project were the two most significant multi-site projects that were running at the organization at the time of our experiment. One of these projects primarily involves the Dutch (NL) site and another site, site A. The other project also involves a third site, site B.

Collaboration between the three sites involved in these two projects differs. In the first project, which involves the NL site and site A, both sites develop distinct software units, and integration of the work done by the sites occurs *between* the software units. In the second project, involving the Dutch site, site A, and site B, collaboration between NL and site B occurs via co-development of the same software unit, whereas collaboration between the Dutch site and site A is organized at the level of unit integration, similar to the first project.

Océ Technologies successfully applies an agile development methodology to encourage creativity and productivity. The organization is flat and workers are encouraged to be proactive in owning up to work responsibilities. Océ Technologies has deliberately opted for cooperation, as opposed to hierarchy, to foster innovation and entrepeneurship.

Table I
PRACTICES FOR ARCHITECTURAL KNOWLEDGE MANAGEMENT IN GLOBAL SOFTWARE DEVELOPMENT

Below, the list of AKM practices for GSD is provided:

- Frequent Interaction Across Sites [10], [12]
- Prioritize Architectural Rules [11]
- Write Down Deviations from Architectural Rules [11]
- Verify Compliance with Architectural Rules [11]
- Cross-Site Delegation (Designated Representatives of Each Team Visit Other Teams) [10], [12]
- Face-to-Face Project Kick-off [10], [12]
- Urgent Request [10], [12]
- Collocated High-Level Architecture Phase [10], [12]
- Clear Project Structure (with Communicating Responsibilities) [10], [12]
- (single) Repository for Architecture Artifacts [10], [12]
- Know Who is Who (Across the Project) [12]
- (single) Configuration Management System for Work Products and Source Code [12]

Agility does not contradict architecture. The agile development process used specifies that the requirement specifications and the architecture design specifications must be approved by the project leaders and the Architecture Council. The lead architect and the unit architects have regular (typically weekly and when needed) meetings to discuss design, and individual teams have their daily meetings.

The agile development methodology that Océ has been using for the past five years is based on SCRUM [13]. SCRUM is an agile software development methodology that allows for adaption for use in a GSD setting (see e.g. [14]–[16]). A software development cycle at Océ typically lasts 8 weeks, with sprints of 2 weeks.

C. Research Approach

From the organization, we selected 38 employees working on the two aforementioned projects: 19 were working at the Dutch site, 14 at site A, and 5 at site B. This distribution is proportional to the number of employees at each of the development sites. The roles and experience of the employees ranges from team leaders to designers, architects, and integration responsibles. The majority of these roles has architecting responsibilities.

On average, the employees had 4.45 years of experience in their current role, with a range between 0.5 - 20 years. The employees were working on average 10.33 years at the organization, with a range between 0.5 - 29 years.

We held interviews with the 38 employees individually of about one hour to one-and-a-half hour. The interviews were held according to a predefined questionnaire that was developed in close cooperation with the organization. The topics of the questionnnaire covered communication and collaboration, knowledge sharing, relationship (and trust) issues, and quality and productivity[1]. In all these topics, various subquestions were formulated to collect information for the validation of our model; questions regarding communication, collaboration, and knowledge sharing focused on how interaction occurs and how knowledge is shared. We video-recorded the interviews to facilitate a thorough analysis afterwards. In a companion article [17], a subset of these interviews is used to study the impact of governance structures on knowledge management.

We performed content analysis on the coded (transcribed) representations of the text to analyze the interview transcripts and validate our research model. Content analysis is a research method that uses a set of procedures to make valid inferences from text [18]. This helps in e.g., revealing the focus of individuals, or groups. We examined broken down pieces of text or semantic units (not single words, but "word sense") from each interview transcript with our research question in mind, and coded answers to the research question using an interactive set of concepts; an interactive set of concepts enables us to code what the practitioners actually do, rather than framing it a priori to our validation model. Since we are interested in a possible distinction between architectural knowledge management practices and the sites involved, we coded for frequency of these concepts, rather than just their existence (termed "conceptual analysis", according to [18]). After we finished coding the 38 interview transcripts, we linked the individual coded fragments to our validation model (the set of practices for AKM in GSD) by inferencing the meaning of the fragments. We performed such a mapping for each of the three sites individually to allow us to compare the results across sites. Intermediate results were discussed in our research team to ensure validity (see Section IV). Fragments that could not be mapped onto the validation model are analyzed separately since these fragments could reveal the existence of other practices in use at the organization that can be leveraged.

III. RESULTS

In this section, we describe the results of our study. The validation of the architectural knowledge management practices for global software development is described in Section III-A. Next, Section III-B provides practices that were found to be performed at, or needed by, the organization, but were not part of our validation model.

[1]See \tt{http://www.cs.vu.nl/~viktor} for more information.

We have structured the results per development site. Because the vast majority of the 38 employees included in our study have architecting responsibilities, we did not further analyze the results to identify differences based their roles. Table II shows the frequency of references made to existence AKM practices for GSD (between brackets the reference to non-existence of that practice).

A. Validation of AKM Practices for GSD

In this section, we discuss the major findings of Table II. We combine the findings of each of the development sites, yet indicate specific similarities or differences observed. Figure 2 provides an excerpt of the codification and aggregation of concepts to our validation model. In this excerpt, various interaction means (e.g. tools) are shown, and annotated with additional reasons for using these means.

Frequent Interaction Across Sites – The organization uses a wide variety of means to ensure sufficiently frequent interaction across sites: periodic meetings (e.g. weekly or biweekly) to share concerns such as planning issues, bottlenecks, and estimates, the use of video-conferencing and Communicator for face-to-face interaction and desktop/code sharing, and of course email and phone communication. Our analysis shows that especially Communicator-tooling for e.g. video-conferencing is greatly appreciated and meant to lower the barriers of distance in collaboration with remote sites; these findings support the findings from Damian et al. [19]. However, despite this variety of means, employees indicate that they do not automatically share the result of these local or informal discussions (e.g. coffee-talk) to other sites; rather, the request for that knowledge should be put forward by the other site. This is inherent in the philosophy of the organization: take responsibility, and ask colleagues when needed. We have identified clear differences in the preference to use either the phone (synchronous communication, a higher sense of urgency transferred) or email to share information or ask questions; some people find themselves more confident in using email because it allows them to verify their writings before actually communicating them (i.e. sending the message). These difficulties lie in language barriers and misinterpretation.

At site A, the need for a single whiteboard for distibuted design meetings (such as described in [20]) was expressed. At site B, we observed that Communicator was used heavily and valued very much by the employees; they mentioned that phone actually was not used that often; in case of an urgent matter, the issue is put forward via email and followed up with a Communicator call to attract attention at the other development site.

Prioritize Architectural Rules, Write Down Deviations, Verify Compliance – These practices are not used by Océ Technologies. Given the philosophy of entrepreneurship and pro-activity as elaborated on in Section II-B, Océ does not pose restrictions such as architectural rules [11] to the employees. Rather, the organization promotes that employees actively request knowledge at the source (i.e., their colleagues) and correct and improve any subobtimal situation they observe along the run.

Cross-Site Delegation – The organization heavily relies on cross-site visits to estabish trust. Several of the key players (project managers and integration managers) at all sites involved in our study have monthly and bi-monthly visits scheduled in their agendas. Although the interview participants mention that it is necessary to prepare the visits well (to get the most out of them), the general opinion is that the benefits outweigh the drawbacks, as supported by [21]. As an example, an employee from site A indicated:

> *"And when you are in the mindset where you just say well, the Dutch people are not right then you're not really building a good relationship. (. . .) And when you really start to go there and to work with them and find out that there are lot of people that really communicate well and you can make friends there as you can make friends here. But this can only happen if you go there. You cannot make friends by email."*

Gaining trust is particularly important for employees working at site B, the site that was added most recently to the organization. The respondents indicate that cross-site visits explicitly were initiated when site B was started five years ago. Currently, however, the respondents indicate that not all employees from site B are eligable for travelling to the other sites; nor are there any regular team-building events to build trust that include employees working at site B, according to the employees working there.

Face-to-Face Project Kick-off – Although this practice is not mentioned very frequently by the interviewees, all sites report on having a face-to-face meeting at the start of a project. Apart from that these meetings help to gain trust early in the project (and thus provide a solid basis for future cooperation across sites), employees from the Dutch site indicate that project kick-offs were conducted at site B when this site was started about five years ago.

Urgent Request – The organization promotes a low threshold for knowledge sharing and coummunication; hence, we observe that the Dutch site often poses and receives questions (either via email or Communicator-chat) that quickly need resolution. This shows that the helpfulness and willingness to share knowledge with the other sites (site A and B) is significantly present. A shared goal exists for each of the two projects (delivering correctly functioning software in time), and an urgent request mechanism clearly contributes to that goal by minimizing knowledge sharing delays.

Collocated Architecture Phase – The practice that promotes the development of the architecture is collocated (i.e., at a centralized location) is not that often mentioned. There

Table II
OVERVIEW OF RESULTS PER SITE. NUMBERS DENOTE THE FREQUENCY OF MENTIONING EXISTENCE OF THE PRACTICE; NUMBERS BETWEEN PARENTHESES DENOTE THE FREQUENCY OF MENTIONING THE NON-EXISTENCE OF THAT PRACTICE

AKM Practice in GSD	Site NL (N = 19)	Site A (N = 14)	Site B (N = 5)
Frequent Interaction Across Sites	54 (13)	38 (5)	22 (1)
Prioritize Architectural Rules	–	–	–
Write Down Deviations from Architectural Rules	–	–	–
Verify Compliance with Architectural Rules	–	–	–
Cross-Site Delegation (i.e. Travelling)	23	9	2 (3)
Face-to-Face Project Kick-off	3	2	2
Urgent Request	13 (1)	1 (1)	–
Collocated (High-Level) Architecture Phase	3	–	–
Clear Project Structure (With Clear Responsibilities)	3 (2)	9	2
(single) Repository for Architecture Artifacts	1 (5)	1 (4)	(1)
Know Who is Who	16	7	1
(single) Configuration Management System	1 (36)	3 (8)	1 (3)

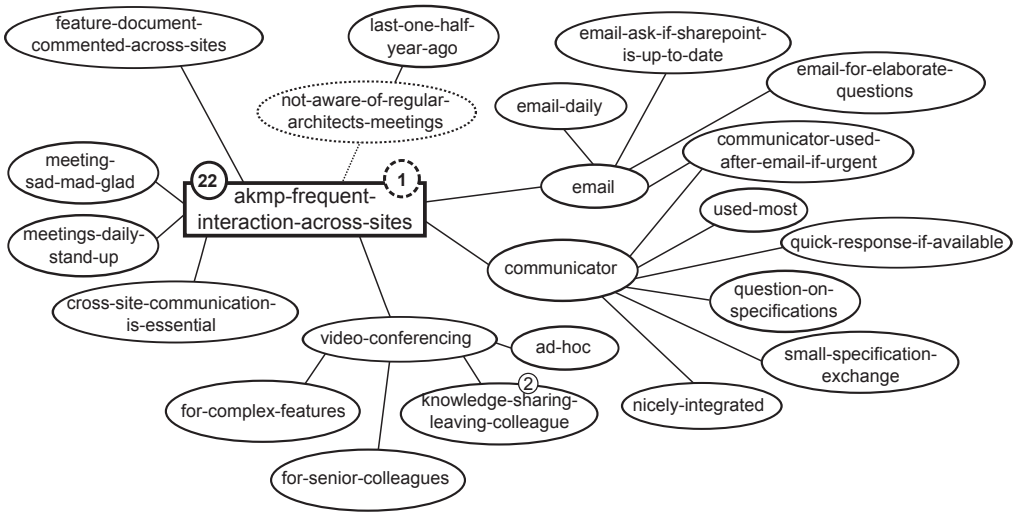

Figure 2. Excerpt of mapping the content analysis results to a part of our validation model (negations shown in dashed lines, the architectural knowledge management practice "frequent interaction across sites", site B).

is an Architecture Council where decisions that cross-cut projects and affect multiple units are taken. Decisions at the level of an individual unit are left to the unit architect – who may reside at another site –, who has regular meetings with the lead architect of the project.

Clear Project Structure – The employees at site A indicate a more hierarchical project organization with clear responsibilities as compared to the Dutch site and site B. As an example, employees from site A indicate that most of the knowledge sharing occurs via (delegation to) the so-called "function responsible" instead of contacting possible knowledge sources directly. According to Hofstede [22], this can be attributed to a difference in the degree to which members of organization accept that power is distributed unequally; in other words, a more "hierarchical structure". The organization perceives a change in the difference in hierarchy between the site A and the Dutch site, as indicated by this example:

> *"Just as an example, when I talk about maybe five, 15, 20 years ago, I was [at site A] and I know people who work in a certain problem; (. . .) then in a coffee break, I asked a developer about the problem, (. . .) which was not very much appreciated by my colleague [at site A] (peer) (. . .) This has changed I think so people are much more lets say equal and its the same flat organisation as it is here."*

(single) Repository for Architecture Artifacts – Architectural documents containing architectural decisions and rationale are not stored in a single repository in this organization. Our analysis reveals various examples in which e.g. employees from site B send an e-mail to ask if the information put onto the SharePoint environment is up-to-date. In addition, in The Netherlands, employees ask their colleagues via email to provide them with an architecture document.

Again, these examples show that the organization relies on the pro-activeness of its employees to initiate and promote knowledge sharing across the development sites, and little on codified knowledge.

Know Who is Who – As with the other sites, it is important for the employees at site A to build a trust network with employees at the other sites. Key to building this network is to "know who is who" and to "know who knows what". Means to achieve this include installing a so-called directory (or "yellow pages") as we have observed earlier in [11] and further elaborated on in [23]. We have not observed any such implementation at this organization, yet the goal (to "know who is who") is deemed important.

(single) Configuration Management System – No shared configuration management system is in place as a knowledge base or for document storage. In the Netherlands, the employees indicate that a myriad of means exist to store information: SharePoint, wikis, a proprietary "document finder" tool, network drives, and email; all are possible locations where information is stored. As a result, artifacts that should be shared (e.g. problem records, meeting minutes) are in fact not shared and employees rely on their colleagues to find out where a certain piece of information is stored:

> *"There is data management on both sides; some [documents] are shared, some not, and some are duplicated. So sometimes it is difficult to find the right information or the most recently updated documentation."*

With respect to the validation of the architectural knowledge management practices for GSD, we observe that practices that are used most frequently focus on the interaction between employees (by knowing who is who), using a variety of communication means, travelling, and the possibility to quickly obtain information when needed. On the other hand, practices that focus on capturing and storing knowledge in shared, single respositories or other systems are not used heavily.

B. Additional Architectural Knowledge Management Practices

While analyzing the survey results, we found practices for the management of architectural knowledge that were performed or mentioned by the organization, but that were not part of our initial model. In this section, we describe these practices.

At all sites, we found that absence of a policy for strategy for knowledge sharing was mentioned. Moreover, some respondents mentioned that more directing management statements and management support towards knowledge sharing would be beneficial.

In addition, at the Dutch site several employees indicated that a cross-site architecture team (the Architecture Council) is in place, including members from site A. The organization promotes a balance in the amount of and frequency in which knowledge is shared across sites. Nevertheless, some employees indicate that there is a imbalance in that the Dutch site sometimes is dominant in e.g. taking architectural decisions and *"(. . .) having the final say on quality (. . .)"*; this imbalance causes difficulties in that employees at other sites feel less involved with the decisions taken. Communication occurs towards other sites, where *"(. . .) [we] hope that they receive it well (. . .)"*. These findings are supported by [24]; the configuration of the development sites and the configuration at each development site individually significantly affects the dynamics of the organization. Furthermore, an imbalance in the size of the teams at each site can even invoke competitive and coalitional characteristics [24].

C. Major Findings

In this section, we provide the major findings of our validation.

Océ Technologies performs several activities that lead us to uncover an additional AKM practice for GSD that can be leveraged for organizations involved in GSD: "peered sites". This practice covers the combination of activities that support a balance in decision-making power: establishing a cross-site architecture team, balancing the frequency of information exchange, and setting and promoting a shared goal for the project activities across development sites. In addition, it is suggested to have more balanced teams in terms of number of employees working at each site.

In addition, we find that Océ puts great emphasis on architectural knowledge management practices for GSD that promote *decentralization*: all sites involved in our study interact frequently with each other without having a predefined top-down communication structure. The key players of a site travel to other project locations to share knowledge, employees voluntarily respond to urgent requests for knowledge from their colleagues, and, finally, project kick-offs are held with members from all sites (physically) present. Practices that place a stronger emphasis on *centralization*, such as establishing a single repository for artifacts or having a single, shared configuration management system are not used; rather, the opposite is true; knowledge and artifacts are stored in multiple locations, causing some difficulties to retrieve the information. Several employees denote a need for having a central repository for artifacts in general and for artifacts that contain architectural knowledge in particular.

The tendency towards decentralization has two major consequences. First, Océ places a great emphasis on practices that support a personalization approach towards knowledge management as opposed to those that support a codification approach towards knowledge management. This corresponds with the nature in which software development activities are undertaken at this organization: in an agile approach, the pro-activeness and entrepreneurship of individual employees promotes practices that focus on collaboration and bringing the knowledge workers together (see [25]), rather than on codifying all knowledge in a single repository. Yet, the employees denote a clear need towards having a repository for artifacts in general and for artifacts that contain architec-

tural knowledge in particular. The myriad of means to store and share knowledge that are in place at the organization result in that several employees do not know where (the latest version of) a document is and whether it is up-to-date or maintained. Following Desouza and Evaristo [26] and Farenhorst et al. [27], a more hybrid knowledge sharing approach combining codification and personalization practices can prove beneficial.

A second consequence of the tendency towards decentralization is that the practices focusing on architectural guidelines that should be adhered to (architectural rules) are not used; posing architectural rules typically occurs in a more centralized collaboration structure. Océ Technologies has deliberately chosen not to pose these rules since the quality mindedness and adherence to quality standards are key responsibilities of each employee individually.

IV. Threats to Validity

In this research, we have validated a collection of architectural knowledge management practices for global software development at a single large organization; we included two projects in our study that involve unit teams spread across three sites. Including only one organization puts some limits on the external validity (generizability) of our research resuls. Yet, having included 38 employees with varying tasks and responsibilities at the three sites, we believe to have included a representative subset of a large organization. Furthermore, the results of this study primarily apply to organizations who are involved in agile-based global software development. Organizations that support a more hierarchical and/or formal software development methodology might benefit from practices that are proposed but are not used by the organization where we conducted our survey. We have used content analysis as the basis of our study. As indicated by [18], several difficulties exist in the application of content analysis that may pose validity issues. We discuss these briefly in the remainder of this section. First, with respect to *measurement*, we decided to count the number of occurrences of the semantic unit. Weber [18] argues that in counting semantic units and subsequently grouping them to an element of our validation model, it requires less effort to identify successive mentions of a word as compared to the first occurrence; however, consistent use in terminology (i.e., Océ terminology) has eased this process. Furthermore, we have also investigated possible reasons for *not* being able to link any semantic unit to our validation model (see Section III-B). Second, with respect to *indication*, we have clearly added `no` or `non` in a semantic unit to indicate the negation of a practice or element (see Figure 2). Since we are interested to use the interview transcripts to answer our research question and not compare interviews or interviewees individually, we have not paid much attention to the use of adjectives or other semantical information (such as understated or overstated words). Third, with respect to *representation* and *interpretation*, we have performed the content analysis of the 38 interviews in a phased approach. This approach allowed us to reflect on possible ambiguities in terms used within our research group.

V. Conclusions

We have conducted a survey at a large agile multi site organization (Océ Technologies) involved in global software development to identify how architectural knowledge actually is shared across development sites. We have analyzed the results of 38 interviews held with employees from three development sites active in two large projects of the organization. Analysis of the results and relating them to our validation model helps to identify the pros and cons of using architectural knowledge and architectural knowledge management practices.

We conclude that architectural knowledge management practices that promote decentralization get much more attention than those promoting centralization at the agile GSD organization. The organization uses practices that focus on interaction and knowledge sharing bottom-up, e.g. between employees individually instead of putting forward (centralized) guidelines and tools. This focus on decentralization leads to an emphasis by Océ on personalization practices (e.g., bringing knowledge workers together) as compared to codification practices.

Finally, we identified one additional useful practice for AKM in GSD: “peered sites”. This practice covers activities that support a balance in decision-making power across sites.

Future work based on this research may include relating the use of the AKM practices for GSD to the quality of the architectures that are developed. This will help in even further analyzing the adequateness of the AKM practices for GSD.

Acknowledgments

This research has been partially sponsored by the Dutch Joint Academic and Commercial Quality Research & Development (Jacquard) program on Software Engineering Research via contract 638.001.406 GRIFFIN: a GRId For inFormatIoN about architectural knowledge and the Dutch “Regeling Kenniswerkers”, project KWR09164, Stephenson: Architecture knowledge sharing practices in software product lines for print systems.

References

[1] B. Al-Ani and D. Redmiles, “Investigating Decision Making Processes in Distributed Development Teams: Findings of a Comparative Empirical Study,” in *Fourth IEEE International Conference on Global Software Engineering (ICGSE’09)*. Limerick, Ireland: IEEE Computer Society, 2009, pp. 51–60.

[2] J. Kotlarsky, I. Oshri, and P. C. van Fenema, *Knowledge Processes in Globally Distributed Contexts*, ser. Technology, Work and Globalization. New York: Palgrave Macmillan, 2008.

[3] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice, second edition*, ser. SEI Series in Software Engineering. Boston: Addison-Wesley Pearson Education, 2003.

[4] M. A. Babar, R. C. de Boer, T. Dingsøyr, and R. Farenhorst, "Architectural Knowledge Management Strategies: Approaches in Research and Industry," in *Second ICSE Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent 2007 (SHARK-ADI'07)*. Minneapolis, MN, USA: IEEE Computer Society, 2007.

[5] I. Rus and M. Lindvall, "Knowledge Management in Software Engineering," *IEEE Software*, vol. 19, no. 3, pp. 26–38, 2002.

[6] GRIFFIN, "Griffin project website," 2011, http://griffin.cs.vu.nl.

[7] R. C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A. Jansen, "Architectural Knowledge: Getting to the Core," in *Third International Conference on the Quality of Software Architectures (QoSA 2007)*, ser. Lecture Notes in Computer Science, S. Overhage, C. Szyperski, R. Reussner, and J. A. Stafford, Eds., vol. 4880. Boston, USA: Springer Berlin / Heidelberg, 2007, pp. 197–214.

[8] J. S. van der Ven, A. Jansen, P. Avgeriou, and D. K. Hammer, "Using Architectural Decisions," in *Supplement to the Proceedings of the Second International Conference on the Quality of Software Architectures (QoSA 2006)*, C. Hofmeister, I. Crnkovic, R. Reussner, and S. Becker, Eds., Universität Karlsruhe, Facultät für Informatik, 2006.

[9] V. Clerc, P. Lago, and H. van Vliet, "The Architect's Mindset," in *Third International Conference on the Quality of Software Architectures (QoSA 2007)*, ser. Lecture Notes in Computer Science, S. Overhage, C. Szyperski, R. Reussner, and J. A. Stafford, Eds., vol. 4880. Boston, USA: Springer Berlin / Heidelberg, 2007, pp. 231–249.

[10] V. Clerc, "Towards Architectural Knowledge Management Practices for Global Software Development," in *Third ICSE Workshop on SHAring and Reusing architectural Knowledge (SHARK'08)*. Leipzig, Germany: ACM, 2008, pp. 23–28.

[11] V. Clerc, P. Lago, and H. van Vliet, "Assessing a Multi-Site Development Organization for Architectural Compliance," in *6th Working IEEE/IFIP Conference on Software Architecture (WICSA 2007)*. Mumbai, India: IEEE Computer Society, 2007.

[12] ——, "The Usefulness of Architectural Knowledge Management Practices in GSD," in *Fourth IEEE International Conference on Global Software Engineering (ICGSE'09)*. Limerick, Ireland: IEEE Computer Society, 2009, pp. 73–82.

[13] K. Schwaber and M. Beedle, *Agile Software Development with SCRUM*, 1st ed. Prentice Hall, 2001.

[14] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Using Scrum in a Globally Distributed Project: A Case Study," *Software Process: Improvement and Practice*, vol. 13, no. 6, pp. 527–544, 2008.

[15] M. Paasivaara and C. Lassenius, "Using Scrum Practices in GSD Projects," in *Agility Across Time and Space*, D. Šmite, N. B. Moe, and P. J. Ågerfalk, Eds. Springer-Verlag, 2010, pp. 259–278.

[16] J. Sutherland, G. Schoonheim, and M. Rijk, "Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams," in *42nd Annual Hawaii International Conference on System Sciences (HICSS'09)*. IEEE Computer Society, 2009, pp. 1–8.

[17] C. Manteli, B. van den Hooff, A. Tang, and H. van Vliet, "The Impact of Multi-site Software Governance on Knowledge Management," in *accepted for Sixth IEEE International Conference on Global Software Engineering (ICGSE'11)*. Helsinki, Finland: IEEE Computer Society, 2011.

[18] R. P. Weber, *Basic Content Analysis, Second Edition*, ser. Quantitative Applications in the Social Sciences. Sage Publications, 1990.

[19] D. Damian, S. Marczak, M. Dascalu, M. Heiss, and A. Liche, "Using a Real-Time Conferencing Tool in Distributed Collaboration: An Experience Report from Siemens IT Solutions and Services," in *Fourth IEEE International Conference on Global Software Engineering (ICGSE'09)*. Limerick, Ireland: IEEE Computer Society, 2009, pp. 239–243.

[20] M. Cataldo, C. Shelton, Y. Choi, Y.-Y. Huang, V. Ramesh, D. Saini, and L.-Y. Wang, "CAMEL: A Tool for Collaborative Distributed Software Design," in *Fourth IEEE International Conference on Global Software Engineering (ICGSE'09)*. Limerick, Ireland: IEEE Computer Society, 2009, pp. 83–92.

[21] C. Sadun, "Scrum and Global Delivery: Pitfalls and Lessons Learned," in *Agility Across Time and Space*, D. Šmite, N. B. Moe, and P. J. Ågerfalk, Eds. Springer-Verlag, 2010, pp. 71–89.

[22] G. Hofstede, *Culture's Consequences: International Differences in Work-Related Values, second edition*. Sage Publications Inc., 2001.

[23] P. Lago, R. Farenhorst, P. Avgeriou, R. C. de Boer, V. Clerc, A. Jansen, and H. van Vliet, "The GRIFFIN Collaborative Virtual Community for Architectural Knowledge Management," in *Collaborative Software Engineering (CoSE)*, I. Mistrík, J. Grundy, A. v. d. Hoek, and J. Whitehead, Eds. Springer Verlag, 2010.

[24] M. B. O'Leary and M. Mortensen, "Go (Con)figure: Subgroups, Imbalance, and Isolates in Geographically Dispersed Teams," *Organization Science*, vol. 21, no. 1, pp. 115–131, 2010.

[25] M. T. Hansen, N. Nohria, and T. Tierney, "What's Your Strategy for Managing Knowledge?" *Harvard Business Review*, vol. 77, no. 2, pp. 106–116, 1999.

[26] K. C. Desouza and R. Evaristo, "Managing Knowledge in Distributed Projects," *Communications of the ACM*, vol. 47, no. 4, pp. 87–90, 2004.

[27] R. Farenhorst, P. Lago, and H. van Vliet, "EAGLE: Effective Tool Support for Sharing Architectural Knowledge," *International Journal of Cooperative Information Systems (IJCIS)*, vol. 16, no. 3/4, pp. 413–437, 2007.