

How do system and enterprise architectures influence knowledge management in software projects? – An exploratory study of six software projects

Carsten Lucke, Markus May, Nane Kratzke, Ulrike Lechner

Institut für Informatik
Universität der Bundeswehr München, Germany
Werner-Heisenberg-Weg 39
85577 Neubiberg
carsten.lucke@unibw.de
markus.may@unibw.de
kratzke@iabg.de
ulrike.lechner@unibw.de

Abstract: This paper covers the influence of system- and enterprise architectures on knowledge management in software development projects. The common impact of architectures is researched in the context of six case studies of medium and large sized software- and system development as well as technical and organizational consultancy companies in the military and non-military domain. Observations are collected in a wide variety of aspects and evaluated on the basis of the Probst et al. knowledge management model [PRR06].

1 Introduction

A lot can happen in a several decades long lifespan of a major information system or enterprise: minor and major changes of its structure, requirements or technology. How to make sure that the knowledge about the requirements, the design decisions, algorithms or processes and the history of changes is sustained such that necessary changes can be made now and in the future? How to prepare for prospective changes?

Usually in complex projects, at some point the majority of the consultants, system designers and engineers who initially built a system or an enterprise architecture will have left the project or even the company. Nevertheless, emerging issues need to be checked and solved as fast as possible. Concerning software systems the arising questions could be: How severe is the reported software problem? Does its underlying cause lie in the software, the system design, the requirements, the handling of the system by users or in the training of the users? Considering enterprise architectures a possible issue could be a changed market situation to which the company needs to respond by establishing a new business strategy: Do certain business processes need to be changed? Is the IT infrastruc-

ture able to cope with intended changes?

These kinds of questions can only be properly answered, if the responsible people have access to knowledge: knowledge about the actual system or enterprise requirements, design decisions, algorithms, documentation and training material and this knowledge needs to be kept alive for decades to be able to respond to those issues. Our research questions are: In how far are system and enterprise architecture used for knowledge management? Who uses them? What kind of support do they provide for knowledge management and how does their usage shape the knowledge management? In how far are they embedded in knowledge management processes and systems?

In our exploratory approach, we present six short cases for usage of architectures in software projects. We collect best practices for system and enterprise architecture usage and the integration of knowledge management and system architectures. We conclude this exploratory study with several hypotheses, based on the identified best practices.

This paper is organized as follows. First, we briefly discuss relevant literature (Sect. 2) and present our method (Sect. 3). Afterwards, we present the six researched cases (Sect. 4), collect the best practices (Sect. 5) and conclude with the presentation of our hypotheses and a brief discussion (Sect. 6 + 7).

2 Theory

The IEEE-1471 standard describes architecture as *fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [Hil00]*. The term architecture is defined heterogeneously throughout literature. Already in the 1990s authors mourned the proliferation of the term »architecture« [Krc90]. When talking about architecture in this paper, we refer to the theory of Armour et al. [AKL99b], depicting architecture as a pyramid: *A software architecture describes the layout of the software modules and the connections and relationships among them. A hardware architecture can describe how the hardware components are organized. However, both these definitions can apply to a single computer, a single information system, or a family of information systems. Thus »architecture« can have a range of meanings, goals, and abstraction levels, depending on who's speaking. An information system architecture typically encompasses an overview of the entire information system – including the software, hardware, and information architectures (the structure of the data that systems will use). In this sense, the information system architecture is a meta-architecture. An enterprise architecture is also a meta-architecture in that it comprises many information systems and their relationships (technical infrastructure). However, because it can also contain other views of an enterprise – including work, function, and information – it is at the highest level in the architecture pyramid. Enterprise architecture is also frequently referred to as system of systems [LS06]. Seen in that way it has to take into account that each system has its own environment of people, sub-*

systems, and data – plus each system must work with other systems to support business operations [AKL99a].

The Zachman framework [Zac87] may possibly be seen as the first major contribution which points out the importance of architectural descriptions as a way to handle increasingly complex (software) systems. Zachman has a strong focus on information systems. However, the proposed framework is nowadays even used as enterprise architecture framework and has inspired many of the currently available frameworks used for enterprise architecture. Most of them have incorporated the use of different perspectives/views to a system, as Zachman suggested. For instance the NATO Architecture Framework (NAF) defines seven views [Cou07] and The Open Group Architecture Framework (TOGAF) provides an architecture vision based on four views [Gro09]. The work of Schekkerman [Sch04a] gives a good overview on a lot of enterprise architecture frameworks and also shows the multiplicity of available solutions.

Several surveys conducted in the recent years show that especially enterprise architecture has grown to a strategic instrument of governance and business alignment [Sch05]. Another approach to standardize architectural descriptions is the IEEE-1471 standard [Hil00]. Maier et al. state that this standard *identifies sound practices to establish a framework and vocabulary for software architecture concepts* [MEH01]. But IEEE-1471 does not only focus on software-intensive systems but also on more general systems like information systems or systems of systems [MEH01]. Lankes et al. find that this standard is even useful for software-cartography of whole landscapes of systems [LMW05]. Referring to Schönherr architecture management and modeling experience a kind of renaissance in the field of enterprise application integration (EAI) since enterprise application frameworks have already been addressing different integration scenarios which bother IT-organizations today [Sch04b].

We observe that architectures are commonly used and broadly accepted in a variety of areas but their benefit to knowledge management, so far is hardly discussed up. In our opinion architectures support knowledge management in organizations, when being properly communicated and made accessible. In this exploratory research work we do neither focus nor constrain on a certain kind of knowledge like tacit or explicit [NT95] but try to gather whatever impact or influence architectures can have on knowledge management. In their reviewing assessment on knowledge management and knowledge management systems (KMS) Alavi and Leidner state that according to Davenport and Prusak [DP98] most knowledge management projects have one of three aims: *(1) to make knowledge visible and show the role of knowledge in an organization, mainly through maps, yellow pages, and hypertext tools; (2) to develop a knowledge-intensive culture by encouraging and aggregating behaviors such as knowledge sharing (as opposed to hoarding) and proactively seeking and offering knowledge; (3) to build a knowledge infrastructure – not only a technical system, but a web of connections among people given space, time, tools, and encouragement to interact and collaborate* [AL01]. We suppose that architectures can especially support (1) by providing access to architectural descriptions, which can possibly improve a variety of knowledge management processes.

3 Method

We follow a case-study approach researching six cases, which is considered a reasonable number of cases for a multi-case research [Eis89]. We inductively build our theory on how architectures can be used to support and improve knowledge management. We then identify best practices which emerge from the cross case analysis. The final results of our exploratory research are hypotheses.

Probst et al. define a model consisting of six core processes relevant for knowledge management [PRR06]: (a) knowledge identification, (b) knowledge acquisition, (c) knowledge development, (d) knowledge distribution, (e) knowledge utilization and (f) knowledge preservation. Since this model provides a practically relevant segmentation of knowledge management processes we use it to guide our research and reflect which processes benefit from the use of system or enterprise architectures.

The cases are compiled from interviews with senior system engineers, architects or project managers. The companies employing our interviewees are assembled from different backgrounds (i.e. business firms, government and defense area) to have a good variation and reach meaningful results. The interviews were semi-structured and the main topics covered in the interviews are the approaches in architecture design and management as well as the use of architecture in the daily work routines and the way architectures are communicated to different audiences. Four cases centre on system architecture and the other two on enterprise architecture. We chose to research these two architectural types since they are the most commonly used ones and to cover a range of differing approaches. Another reason is the probability to draw conclusions from the findings concerning one architectural type to the other one. All interviews were conducted in May and June 2008 in the context of a diploma thesis [May08]. We use the interview transcriptions and findings of this diploma thesis as the foundation of this research paper.

4 Cases

Our case studies cover projects in the following six organizations: Bayerische Motoren Werke Group (BMW), European Aeronautic Defence and Space Company (EADS), Industrieanlagen-Betriebsgesellschaft mbH (IABG), International Business Machines Corporation (IBM), Thales Group and the Taktikzentrum Marine (TZM) of the Marineoperationsschule (MOS) Bremerhaven. For anonymization purposes, these organizations are referred to as »Company A-F«, whereas the alphabetical order of these names does not allow for a reverse mapping.

Table 1 shows a summary of the effects of the companies' architecture on the knowledge management processes defined by Probst et al. [PRR06]. A tilde character (~) indicates no particular support, (+) means that a supportive character is perceived and (++) indicates significant support of the regarding knowledge management process.

Knowledge Management Process	Company					
	A	B	C	D	E	F
<i>knowledge identification</i>	++	+	++	~	++	+
<i>knowledge acquisition</i>	~	~	+	~	+	~
<i>knowledge development</i>	+	+	+	+	+	+
<i>knowledge distribution</i>	+	~	+	+	++	~
<i>knowledge utilization</i>	+	+	+	+	+	+
<i>knowledge preservation</i>	~	+	+	+	+	+

Table 1: Overview on how the architecture supports knowledge management processes

Table 2 gives an overview about the target audiences and the degree of utilization of the architecture for knowledge management. The tilde character (~) means no particular use of the architecture, (+) indicates noticeable but irregular use and (++) signalizes significant use of the architecture for knowledge management purposes. The audience classifications are *Manager* (project leader and project manager), *Architect* and *Developer* (with the architect being a chief developer), *Quality Assurance (QA) person* and *End-User*. Concerning system architectures the end-user is a person who uses the developed information system. Regarding enterprise architectures the end-user is a collective term that stands for the stakeholders defined in the enterprise architecture framework.

Target Audience	Company					
	A	B	C	D	E	F
<i>Manager</i>	+	+	++	~	++	+
<i>Architect</i>		++	++		++	++
<i>Developer</i>	++	++	++	+	++	+
<i>Quality Assurance person</i>	+	~	+	~	++	+
<i>End-User</i>	+	~	++	~	++	~

Table 2: Overview about target audiences of the architecture¹

Subsequently follows a more detailed description of the researched cases. We focus on the description of those aspects where architectures have shown to be supportive (+) or significantly supportive (++) for certain knowledge management processes (compare Table 1) or have noticeably (+) or significantly (++) been used by a target audience for knowledge management purposes (compare Table 2).

¹The role of architects was not explicitly mentioned in the interviews with Company A and D, thus the table-cells are left blank intentionally. It can be assumed that the value evaluated for developers applies to architects as well.

4.1 Company A

The project in Company A developed a proprietary multi-tier system architecture. This had a negative impact on the knowledge management processes of »knowledge utilization« and »knowledge acquisition« identified by Probst et al. [PRR06]. First, it was difficult to find employees being already familiar with this architecture and second, new project members generally needed some time to familiarize themselves with this kind of architecture. If the company would have adopted widely known architectures (i.e. open-source architectures) the probability of new project members knowing these would have been higher.

Furthermore, Company A maintained two different perspectives to the system architecture – one of these being data-related and another one providing a functional view to the system. The benefit of having these architectural descriptions was the development of a common vocabulary, used within communications between team members (management, developers and QA personnel). There was no dedicated perspective for the customer’s specialist division or end-users. Concerning project lead and project management the functional perspective was used, to define project teams and also documentation artifacts and configuration management were structured in regard to the functional perspective of the system architecture. This enabled project members having certain knowledge about the system architecture (mostly developers), to find relevant contact persons within the whole team, find relevant pieces of documentation in shorter time and navigate the configuration management system in a more efficient manner.

The software development process adopted by Company A was aligned according to the V-Model 97 [DW99] process model. In this regard, the biggest ascertainable advantage according to knowledge management was, that the process model postulated a certain amount of documentation artifacts to be created.

4.2 Company B

The case study of Company B explored a project concerned with system architecture creation. An external service provider developed the architecture for Company B. The specialist division of Company B was not interested in certain aspects of the architecture but only in the business needs that the final system was designed to fulfill as well as integration aspects of the system into the whole IT landscape. As a result the project maintained no dedicated perspective for employees of Company B. According to Zachman, a »model of the business (i.e. owner’s view)« would be a good perspective [Zac87] on how an information system works and it is well targeted at the audience of a business-specialist division. In this case though, no such perspective was documented. However, the external service-provider (which includes developers and architects) gained much advantage from properly documenting the scope of the architecture’s »technology model« as well as »out-of-context views« [Zac87]: The technology model includes data and process descriptions

and our interview partner mentioned that this model fostered a common vocabulary, used by the external service-provider's employees (managers, developers and architects). Various graphical user interface prototypes (out-of-context views) were used to support the communication with business-specialists as well as end-users.

It attracted our attention that project members of Company B structured their documentation artifacts in regard to the project stages (also in terms of directory structures in connection to file storage) whereas the service provider, who developed the information system architecture, organized its documentation deliverables in regard to the components of the information system architecture. In this regard, the architecture helped managers, architects, developers and QA personnel to structure documentation artifacts. Considering the Probst dimension of »knowledge identification« [PRR06] this means, the architecture adds certain value to identify relevant documentation artifacts. Having the team- and document structure aligned in regard to the architecture's components, helps to find contact persons or relevant documents. This way, it got possible to tell which documents needed to be updated, when specific modules of the system architecture were changed.

4.3 Company C

The project of Company C was about developing an enterprise architecture for a customer. Subsequently we use the term service provider synonymously when we refer to Company C. The enterprise architecture framework to be used – the NATO Architecture Framework (NAF) [Cou07] – was determined by the customer. The NAF defines several views to the architecture: e.g. an operational one and a system view. The operational view provides information about the organization's operational nodes, their tasks and how they are connected to each other. The system view illustrates the functionalities of the customer's organizational units. Describing the enterprise architecture with this set of different perspectives, allowed employees of the customer with different knowledge or backgrounds to get an easier and better understanding of the described circumstances.

In this case, no dedicated process model was used in the design process of the enterprise architecture model. Thus, the process model could not act as a catalyst for creation of documentation artifacts. However, this had no negative effect on the knowledge management since the enterprise architecture framework suggested a development method and defined certain architectural descriptions (views) to be created.

The process of analyzing the customer's current organizational architecture and the definition of the target enterprise architecture aligned to the business goals was active knowledge management. The employees of both, the customer and the service provider got deeply involved with the current enterprise architecture which allowed them to identify weaknesses (i.e. redundancies or disharmonies) and derive necessary changes for the architecture to be defined.

Mapping these facts to the Probst knowledge management model [PRR06], the architecture fosters a very effective »knowledge identification« which in turn directly supports »knowledge acquisition« and »knowledge development«. The creation and application of a clearly defined architectural model resulted in the establishment of a common vocabulary on both sides, customer and service-provider. This enhanced and alleviated communication inside and across teams. For this reasons, all target audiences (management, end-users, QA personnel as well as architects and developers) were assigned a (+) or (++) rating in table 2.

As a final result of the project, Company C defined and established an enterprise architecture, which could be presented to the customer's employees in different views. These different models were enriched with documentation artifacts providing detailed information. Hyperlinking all these elements allowed for an efficient navigation and combined a good overall overview with easy comprehensibility. Thus, the aspect of »knowledge identification« [PRR06] is strongly supported by the established enterprise architecture.

4.4 Company D

In the case of Company D a proprietary system architecture was developed, based on established architectural styles like client/server and peer-to-peer. While the usage of a proprietary architecture lowered the possibility of finding staff being familiar with it, the well known architectural styles partly compensated this downside and reduced the amount of knowledge new members of the project had to acquire.

The company maintained only a single perspective to the system architecture which was – referring to Zachman [Zac87] – a »technology model« providing »data- and process descriptions« of the system. This lack of different perspectives to the system architecture excluded project members, apart from developers and architects, from getting acquainted with the architecture. It is perceived to be unfortunate that even the model's target audience (i.e. software engineers and -architects) did not capaciously use the model, as it was outdated since changes to the physical architecture had not been incorporated in the documents. The communication about different aspects of the software system within specialized teams (e.g. developer to developer communication) worked rather well. But it was apparent, that as soon as communication between project teams or team members with different backgrounds (e.g. software-engineer to end-user communication) was required, the lack of a shared vocabulary or awareness of the system in general prevented those team members from communicating efficiently.

4.5 Company E

In the case of Company E an enterprise architecture was developed for a customer, based on the TOGAF framework [Gro09]. The framework defines four architectural domains:

business architecture, application architecture, data architecture and technical architecture. Company E added the dimension of »capability architecture« to the framework. The purpose of this additional dimension was to allow for a periodically conducted analysis of the companies capabilities.

The goal of the creation-process of the new enterprise architecture was to provide all knowledge regarding the five dimensions (e.g. detailed information about business units, IT infrastructure landscape, etc.) within an architecture database, to make information searchable. Several facts are apparent in this case: Already the analysis of the customer's current enterprise architecture adds major value in terms of »knowledge identification«. Since members of all audiences listed in table 2 were involved in this process they all benefited in a positive way. The creation of an architecture database containing valuable information about the enterprise architecture combined with powerful search functionalities helps to identify relevant knowledge and at the same time »knowledge distribution« gets a significant boost. Knowledge retrieval was changed from a »push« to a »pull« principle which reduced information overload. Information inside this architecture database was kept up-to-date with the support of specific tools.

The »knowledge utilization« dimension defined by Probst et al. [PRR06] is supported by providing information in different perspectives, each adjusted to a specific target audience. This supports ease-of-use of the available information. Employees on a strategic level (i.e. management) for instance, get an overview about the enterprise as a whole and processes whereas IT personnel can view information from an application- or technical infrastructure centric perspective.

Last but not least »knowledge acquisition« and »knowledge development« are improved by introducing the additional architectural domain of abilities, which allows for a periodical analysis on whether knowledge needs to be acquired externally or developed internally in order to accomplish enterprise goals.

4.6 Company F

The fundamental observation in the case concerned with Company F is, that lack of time resulted in a significant shortage of documentation. Focus of the project was the development of a client/server-oriented system architecture for a customer, using a proprietary architecture framework. Nevertheless, the fact that the well-known client/server architecture model was used, improved chances, that new project members (in most cases software engineers) were familiar with the very basics of the system architecture. Thus, considering the Probst et al. dimension of »knowledge acquisition« [PRR06] it is helpful to stick to non-proprietary products.

Since only a »technology model« perspective of the system architecture was maintained, the model was neither used for communication towards the customer, nor did it enhance

the building of a common vocabulary that could be used in conversations between employees of Company F and the customer. The customer was only interested in the fulfillment of its requirements and showed no interest in the architecture. As described in previous cases, team structure and also the structure and outlining of documentation artifacts were aligned to the technology perspective of the system architecture. Having a certain knowledge of the system architecture, this eased »knowledge identification« (i.e. finding appropriate contact persons or finding relevant documents).

The project employed the V-Model 97 [DW99] process model during the software development process. This process model defined necessary documentation artifacts to be created and thus did add a certain value in terms of »knowledge preservation«.

5 Best Practices

The six cases cover four companies developing system architectures and two companies developing enterprise architectures. Several best practices are gathered from these cases. One obvious best practice is to *incorporate a well-known architecture* rather than a not well-known one (e.g. a proprietary architecture) to support »knowledge utilization«. This best practice is independent from the type of architecture being created – enterprise or system architecture. Companies A, D and F dealt with proprietary system architectures which made »knowledge acquisition« and »knowledge usage« difficult. Still, Company D and Company F had an advantage over Company A by using well-known technology models like client/server or peer-2-peer in their system architecture framework. In these companies new project members were at least familiar with the underlying technologies of the architecture.

Another best practice is to *define architecture-models, showing the architecture from different perspectives*, each targeted at a specific audience. Both, »knowledge utilization« and »knowledge development« benefit from this practice. The cases showed, that independent from which target audience is addressed, an architectural model is rather used, if it is understandable for the specific group of people, than a model, which is too detailed or even outdated (Company D). Whenever a model is accepted and used by the target audience, it helps to establish a common vocabulary, which makes communication easier and more efficient (Company A, B, C and F). Cases concerned with system architecture development typically used architectural models and related vocabulary for communication within development teams, but not for communication towards end-users or customers (Company A, B and F). By contrast, all cases concerned with enterprise architecture, developed much more architectural models to give a view on the architecture from several different perspectives. In each of these cases, this resulted in the establishment of a commonly used vocabulary, enhancing communication within teams and across team- or division-boundaries (Company C and E).

One important lesson learned from the cases researching system architecture is to *structure*

documentation artifacts, configuration management systems (e.g. bug-tracking systems, source-code management systems) or even teams in regard to the architecture. Project members, having a basic knowledge about the architecture, benefit from this by finding relevant information or contact persons in an efficient manner (Company A, B and F). This practice supports the »knowledge identification« dimension identified by Probst et al. [PRR06]. This structuring helped Company B in terms of »knowledge preservation«, since whenever changes were done to the physical architecture it was obvious which documentation artifacts needed to be updated.

The cases, which dealt with enterprise architectures (Company C and E), did also focus on the structuring of knowledge management artifacts (e.g. documents, contact person information) in alignment to the architecture. Even better, beyond providing a plain architectural description, they did *use the architecture model(s) as interactive index of contents*. We call this a knowledge map. Models, depicting structural components of the enterprise architecture, are made accessible to employees in different views, showing the architecture from different perspectives. This gives »knowledge identification« as well as »knowledge utilization« a significant boost. Furthermore, Company C enriched these models by hyper-linking documents or other knowledge management artifacts, providing detailed context information about certain nodes in these architectural models.

6 Research Hypotheses

This section of the paper has a reflective look at a selection of our most important hypotheses and determines whether the observed results corroborate them.

1. *Both, system and enterprise architecture have an influence on the knowledge management processes defined by Probst et al. [PRR06].*

This assumption proves to be true in several aspects and the influence can be positive as well as negative. Company A for instance used a proprietary system architecture framework and had difficulties finding new project members, being already familiar with the architecture. Thus, the architecture has a negative influence on »knowledge acquisition«. On the other hand the cases show that architectural descriptions can support a number of knowledge management processes like for instance »knowledge identification« (Company C and E), »knowledge distribution« (Company E) or »knowledge preservation« (Company B).

2. *The use of an architecture leads to the establishment of a common vocabulary and therefore improves communication between team members.*

In all case studies, a common vocabulary is introduced by having an architecture or more precise by having one or more architectural descriptions. System architectures mostly help to build a common vocabulary within teams (Company A, B, D and F), whereas enterprise architectures help to introduce a vocabulary used within

teams as well as across team boundaries (Company C and E). An ideal architecture offers an architectural description to each target audience it intends to address. Enterprise architecture seems to be very effective in reaching its intended target audiences (see table 2). Certainly the reason for this is, that an architectural description is developed for each of the stakeholders and the defined target audience does not go beyond this group of people. However, typically a company consists of many more people but those stakeholders. We believe that extending the use of an enterprise architecture to a wider audience would be to the further benefit of a company's knowledge management processes. In any case, the architectural descriptions help to make communication more efficient and avoid misunderstandings.

3. *An architecture will not be used by people who do not understand the architectural descriptions.*

This hypothesis is closely connected to hypothesis (2). Most cases, concerned with both, system and enterprise architecture, provided more than a single perspective/view to the architecture. In the case of Company B for instance, the external service provider used a »technology model« [Zac87] to enhance the communication between software engineers (inside-team communication) and »out-of-context views« [Zac87] to enable and improve the communication between software engineers and business specialists or end-users (cross-team communication). On the contrary, Company A did not provide an architectural description suited to the business specialist division of the customer. In the result the architecture was not used in communications to the customer (across-team communication). Thus, it did only enhance the communication within the team of Company A. This tells us, that providing more perspectives/views to an architecture, means getting through to a wider target audience. In this regard architecture frameworks need to find a reasonable number of architectural descriptions.

4. *Architectural descriptions are perfectly suited to be used as knowledge maps and will as such improve several of the knowledge management processes defined by Probst et al. [PRR06].*

Architectural descriptions of both, system and enterprise architectures helped new project members to get an easier overview about the system or company. This helps in terms of »knowledge identification« and »knowledge development« because people are able to find relevant documents by using the architectural descriptions as a guiding map. In the cases of Company C and E this idea of using the architectural descriptions as a knowledge map was pushed even further. They used the architecture models as an interactive information asset base where context related documents were hyperlinked to provide easy and direct access. This alignment of knowledge management to the architecture results in a major improvement of »knowledge distribution«.

7 Concluding remarks

With our research and this paper we develop theses on how architectures influence knowledge management in software projects. According to our literature review it is pretty much the first research project with this specific scope of examination. The conducted number of case studies provides a solid base for our hypotheses but it is still a first exploratory study in that field. As of today, system architectures are mostly used as an instrument to handle the complexity of software systems and enterprise architectures are used as an instrument of governance and business- or IT alignment to safeguard running investments, plan future investments and furthermore to simultaneously reduce costs [LMW05]. In this respect, architectures are already used for knowledge management but in our opinion their potential is not fully tapped on a broad basis.

We can summarize that architectures are a valuable appliance to provide access to explicit knowledge [NT95] of any kind (e.g. information about system requirements, design decisions or algorithms). Our study shows that architectures support a variety of knowledge management processes, if used as knowledge map – especially »knowledge identification« and »knowledge distribution«. They can also vastly improve inner- and inter-team communication by establishing a common vocabulary.

Further investigations will be the definition of blueprints for architectures that are especially suited to support the knowledge management in software projects (or even in general). We also plan to investigate, how specific target audiences can be reached best and how to communicate an architecture to these audiences. In our introduction we also mentioned, that in complex systems, knowledge often needs to be kept alive and accessible for decades. In this regard, historiography is an important aspect. Further research work could deal with strategies how this can be done directly inside architecture models.

8 Acknowledgements

We applied the FLAE norm [THR⁺07] for the sequence of authors of this paper. The credit for a huge amount of the research work goes to Markus May who conducted and evaluated the interviews of the mentioned case studies. Last but not least, we especially thank all of our interview partners for taking time to talk to us.

References

- [AKL99a] F Armour, SH Kaisler, and S Liu. Building an enterprise architecture step by step. *IT Professional*, 1(4):31 – 39, Jul 1999.
- [AKL99b] F Armour, SH Kaisler, and SY Liu. A big-picture look at enterprise architectures. *IT Professional*, 1(1):35–42, 1999.

- [AL01] M Alavi and DE Leidner. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1):107–136, 2001.
- [Cou07] North Atlantic Council. NATO Architecture Framework (NAF) Ver 3. 2007.
- [DP98] TH Davenport and L Prusak. Working Knowledge: How organizations manage what they know. *Mcgraw-Hill Professional*, (1), Jan 1998.
- [DW99] W Dröschel and M Wiemers. *Das V-Modell 97: Der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz*. Oldenbourg, Nov 1999.
- [Eis89] K Eisenhardt. Building theories from case study research. *Academy of management review*, Jan 1989.
- [Gro09] The Open Group. *Togaf Version 9 - A Manual*. Van Haren Publishing, Jan 2009.
- [Hil00] R Hilliard. IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE*, <http://standards.ieee.org>, 2000.
- [Krc90] H Krcmar. Bedeutung und Ziele von Informationssystemarchitekturen. *Wirtschaftsinformatik*, 32(5):395–402, 1990.
- [LMW05] J Lankes, F Matthes, and A Wittenburg. Architekturbeschreibung von Anwendungslandschaften: Softwarekartographie und IEEE Std 1471-2000. *Software Engineering*, pages 43–54, 2005.
- [LS06] S Leuchter and R Schönbein. Die Verwendung von Architectural Frameworks als Vorgehensmodell für die System-of-Systems-Entwicklung. *INFORMATIK 2006. Informatik für Menschen. Beiträge der 36. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, 2006.
- [May08] M May. Architekturen - Wissensmanagement in IT-Projekten. *Diplomarbeit, Universität der Bundeswehr München*, Jul 2008.
- [MEH01] MW Maier, D Emery, and R Hilliard. Software architecture: Introducing IEEE standard 1471. *Computer*, 34(4):107–109, 2001.
- [NT95] I Nonaka and H Takeuchi. The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. *Oxford University Press*, Jan 1995.
- [PRR06] G Probst, S Raub, and K Romhardt. Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen. *Gabler*, (5. Auflage), Jan 2006.
- [Sch04a] J Schekkerman. How to survive in the jungle of enterprise architecture frameworks: creating or choosing an Enterprise Architecture Framework. *Ebookslib*, Jan 2004.
- [Sch04b] M Schönherr. Enterprise Architecture Frameworks. *Enterprise Application Integration-Serviceorientierung und nachhaltige Architekturen*. Hrsg. S. Aier/M. Schönherr, Berlin: Gito. Reihe: Enterprise Architecture, 2:3–48, 2004.
- [Sch05] J Schekkerman. Trends in Enterprise Architecture 2005: How are Organizations Progressing? *Institute For Enterprise Architecture Developments, Report of the Third Measurement*, Dec 2005.
- [THR⁺07] T Tschardtke, ME Hochberg, TA Rand, VH Resh, and J Krauss. Author sequence and credit for contributions in multiauthored publications. *PLoS Biol*, 5(1):e18, Jan 2007.
- [Zac87] J Zachman. A framework for information systems architecture. *IBM systems journal*, 26(3), Jan 1987.