Faculty of Sciences
Department of Computer Science
Section Information Management and Software Engineering

Technical Report IR-IMSE-003
April 2009

What Architects Do and What They Need to Share Knowledge

Rik Farenhorst
Johan F. Hoorn
Patricia Lago
Hans van Vliet

VU University Amsterdam

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.

Contents

Abstract

Much has been written on software architecture theory and practice. Many of us are familiar to existing best practices on describing architectures, methods to evaluate these architectures, and process guidelines to ensure that architecture development blends in nicely in the software development process. Over the past few years, more attention is put on the role of decision making in the architecting process. Consequently, the role of the architect in this process is a frequently recurring topic of discussion, and more and more researchers and practitioners deliberate on what a proper set of duties, skills and knowledge of architects would be. In addition, researchers have proposed various tools to support the architect, who is characterized as an all-round knowledge worker. Such a knowledge worker, as these researchers argue, would definitely benefit from specific support for managing design decisions, modeling architectural solutions, or related activities that can be automated.

In this report we assess to what extent the above claims hold. The best way to do this is to unravel what architects really do and what they really need to share architectural knowledge. For this purpose, we have constructed a theory on architecting. This theory is based on a state of the art literature review and experiences from a few years of case study research in an actual architecting process. Our theoretical framework consists of a number of identified architecting activities and a number of support methods that assist architects in sharing architectural knowledge during these activities. Further, we hypothesize that a number of 'patterns' exist between these activities and methods: certain methods for sharing architectural knowledge are more supportive than others to an architect involved in a specific activity.

To validate our theory we have conducted a large-scale survey in three Dutch software development organizations. Almost 300 practicing architects were asked about their daily activities, and how important they consider the various types of support for sharing architectural knowledge. The results of our study indicate that architects can be characterized as rather lonesome decision makers who mainly consume architectural knowledge, but neglect to document and share such knowledge themselves. In this technical report we show how respecting this nature of architects is a step towards development of more effective support for architectural knowledge sharing.

Chapter 1

Introduction

Much has been written on software architecture theory and practice. Many of us are familiar to existing best practices on describing architectures, methods to evaluate these architectures, and process guidelines to ensure that architecture development blends in nicely in the software development process.

One of the core processes in software architecting is decision making. In this decision-making process the architects constantly need to balance all kinds of constraints, such as requirements and needs of the customer(s) and other stakeholders, technological and organizational limitations, existing expertise and experience, as well as organizational or architectural best-practices and standards. Based on sound judgment, the architects are expected to take the best architectural design decisions and to communicate and motivate these decisions.

Over the past few years, an increasing attention is put on the role of decision making in the software architecting process [25, 31, 39]. Consequently, the role of the architect in this process is a frequently recurring topic of discussion, and more and more researchers and practitioners deliberate on what a proper set of duties, skills and knowledge of architects would be [6]. In addition, researchers have proposed various tools to support the architect [1, 4, 13, 24], who is characterized as an all-round knowledge worker. Such a knowledge worker, as these researchers argue, would definitely benefit from specific support for managing design decisions, modeling architectural solutions, or related activities that can be automated. With such support relevant architectural knowledge is easier to share, which leads to more effective knowledge exchange and reuse, and which prevents knowledge vaporization [3].

Architects often reuse existing assets such as architectural patterns and styles. They negotiate with other stakeholders, or trust on their 'gut feeling'. Sharing support is crucial to prevent loss of architectural knowledge, to exchange experiences and ideas with colleagues, to (re)use architectural expertise and best practices, and to train junior employees.

There are various approaches to support architects in sharing architectural knowledge. In practice, however, the adoption of these approaches is limited, perhaps due to a lack of alignment to the architecting process. Research has shown that these approaches are often developed from a technological perspective alone [13]. To keep from falling into this so-called ICT trap [22] we wish to find out what support for architectural knowledge sharing best fits the architecting process. However, the field lacks clear insight into the connection between architecting activities and architectural knowledge sharing support.

The only way to understand what would really benefit architects, is to grasp what architects do and what kind of support they need during their main activities. Our main research question thus is:

MRQ1 How can we align support for sharing architectural knowledge with the various architecting activities so that architects can be supported in these activities?

The main research question can be decomposed into what exactly architects do and what kind of support they need. This translates into four detailed research questions:

RQ1 With which activity do architects agree the most as an illustration of their work? In other words, what is their core activity?

RQ2 With which kind of support do architects agree the most as a contribution to their work? In other words, what is their core need?

RQ3 According to the architects, which support is important for each activity? In other words, can we identify patterns for architectural knowledge sharing that define good combinations of activities and support?

RQ4 What do architects consider the best tradeoffs between conflicting methods for architectural knowledge sharing support?

We have constructed a 'theory on architecting', based on a review of state-of-the-art literature on software architecture and knowledge management, and experiences gained during four years of case study research in a large software development organization [15]. Our theoretical framework consists of a number of identified architecting activities and a number of support methods that assist architects in sharing architectural knowledge during these activities. Further, we hypothesize that a number of 'patterns' exists between these activities and methods: certain methods for sharing architectural knowledge are more supportive than others to architects involved in a specific activity. The identification of such patterns would help in the development and maturation of architectural knowledge sharing methods and tools. In addition, the results of this study may also provide us with more focus in the architectural knowledge tools we develop, such as a portal or a wiki [14, 16]. Based on the results of the present study we can determine which specific functionality is appreciated in which particular situation. This information allows us to prioritize development efforts and to tailor future versions of our tools to the architects' needs.

To validate our theory we have conducted a large-scale survey research in four IT organizations in the Netherlands. In total 279 practicing architects were asked about their daily activities, and how important they consider the various types of support for sharing architectural knowledge. Our study consisted of a pilot study to pretest the theoretical framework, followed by a main study to answer our research questions and hypotheses.

Our survey results show that architects are complicated individuals. They make lots of architectural decisions, but neglect documenting them. Codifying and subsequently sharing architectural knowledge is clearly not one of their most popular activities. While in itself not very surprising, one would expect that support to assist architects from this latter task is something they appreciate. The opposite is true. Architects rather stay in control and are not interested in automated or intelligent support. When it comes to consumption of architectural knowledge, however, support for effective retrieval of (stored) architectural knowledge is on the top of their wish list. This apparent contradiction suggests architects are rather lonesome decision makers who prefer to spend their working life in splendid isolation.

The remainder of this report is organized as follows. In Chapter 2, our theoretical framework is elaborated by discussing architecting activities and requirements for architectural knowledge sharing support. In Chapter 3 we outline our research methodology used in order to validate our theory and answer our research questions. Chapter 4 presents a detailed step-by-step overview of our data analysis. In Chapter 5 we discuss the main results and reflect on them. Chapter 6 concludes this report.

Chapter 2

Theoretical Framework

In a recent paper on sharing and reusing architectural knowledge it is argued that architectural knowledge consists of all the knowledge used and produced during architecting [33]. Such architectural knowledge encompasses knowledge of the problem domain (e.g., architectural requirements, drivers, constraints), the solution domain (e.g., architectural tactics, patterns, styles), and also knowledge entities used in the architecting – or decision-making process – itself, such as design decisions, rationale, and the resulting architecture design [12]. The broad scope of architectural knowledge calls for knowledge sharing support that is equally broad in scope.

Based on experiences in earlier industrial case studies, together with established literature on software architecture and knowledge management, we are able to characterize what architects do and what they need to share architectural knowledge. In the next three sections we outline our theory on what kind of activities architects are involved in (Section 2.1), what kind of support they need (Section 2.2), and what patterns between support and activities exist (Section 2.3). An earlier version of this theory has been published [15]. Based on the pilot study (cf. Section 3.1) we improved this theory by making the various categories more orthogonal and better scoped.

2.1 Architecting activities

According to Philippe Kruchten, software architects should "make design choices, validate them, and capture them in various architecture related artifacts" [29]. Doing all these things involves a lot of consensus *decision making* in which architects balance between quality criteria, stakeholder concerns, and requirements, and in which they try to apply architectural styles and patterns.

A study on the duty, knowledge, and skills of architects confirms the above view of the primary tasks of architects [6]. The authors found that architects frequently interact with stakeholders, are involved in organization and business related issues, but primarily guide the architecting process: "We realized, from our many interactions with practicing architects, that their job is far more complex than just making technical decisions, although clearly that remains their most essential single duty" [6].

In the decision making process architects *communicate* and share ideas with various stakeholders and collaborate in teams to find the optimal solution [13]. To keep track of all knowledge being created or shared in this process, architects maintain a backlog [20]. In this backlog, an overview of decisions, constraints, concerns, open issues etc. are maintained by the architect.

Input to the decision-making process is the vast body of architectural knowledge architects need to take the design decisions. According to Clements et al. this body of knowledge is a combination of basic computer science knowledge, plus architectural knowledge in terms of knowledge of technologies and platforms and knowledge about organizational context and management [6]. Architects share this knowledge not only by talking to colleagues in their team, but also by communicating with stakeholders outside the team.

That architecting is not only about technology is also illustrated by Bredemeyer's architect competency list [9]. These competencies are classified in the categories technology, consulting, strategy, organizational politics, and leadership. Bredemeyer's vision is further underlined by Peter Eeles [10], who acknowledges that architects do not only do technical stuff ('they are a technical leader', 'they understand the development process') but that they also need a lot of soft skills: Architects 'have knowledge of the business domain', 'are good communicators', 'make decisions', 'are aware of organizational politics', and 'need to be effective negotiators'. Soft skills are also emphasized by Van der Raadt et al. [40], who studied how different stakeholders in the Enterprise Architecture domain perceive the enterprise architecture function. Apart from technical knowledge or skills, enterprise architecture involves to a large extent governance, communication, vision, and collaboration [40].

Another main architecting activity is to *document* architectural knowledge. This not only includes storing application-generic knowledge such as architectural principles, patterns or styles, but also codifying application-specific knowledge such as design decisions made and their rationale. Architectural knowledge may be stored in architectural descriptions, databases, modeled in reference architectures or grouped in specific repositories.

During the 2006 and 2007 workshops on sharing and reusing architectural knowledge (SHARK) an explicit distinction was made between architecting as a product and as a process [2, 32]. During these sessions a classification scheme for architecting as a process was constructed. This scheme adds two important architecting activities to the decision making, documenting and communication activities: learning and assessing. With respect to learning, Clements et al. list university or industrial courses, and certification programs as possible sources of training and education for architects [6]. Apart from these specific learning activities, in their everyday work architects are also constantly busy with *knowledge acquisition* activities to update their knowledge. By having discussions with colleagues or customers, reading fora on the internet, or by visiting seminars, workshops or conferences they stay up-to-date and become acquainted with new trends, developments or best practices.

The last main architecting activity we define is *assessing the quality* of architectures. Assessing and reviewing architecture descriptions or related deliverables allow architects to control quality. They are ultimately responsible for the quality of the software systems. The design and evolution of an architecture allows them to reach this goal. As part of the quality control, architects also prescribe certain rules and regulations to stakeholders in the projects they work on. Stakeholders need to adhere to these rules, and architects may reject deviations from these rules by overruling specific decisions made.

We collected the main architecting activities in Table 2.1: communication, decision making, quality assessment, documentation and knowledge acquisition. For each of the five activities we defined several subactivities, each of which was used as an item in our survey. Table 2.1 shows the activities and the items that should be most indicative for these activities, which were tested by our online survey.

A1: Communication - I inform colleagues about the results of my work. - My colleagues keep me up-to-date on the results of their work. - I explain existing architectural principles to colleagues. - During discussions I share my knowledge to colleagues.
A2: Decision Making - Before I take a decision I weigh the pros and cons of possible solutions. - While taking decisions I meet the wishes of the stakeholders. - I think about what impact my decisions have on the current architecture. - I study the reasoning behind taken design decisions.
A3: Quality Assessment - I convince stakeholders in order to share the architectural vision. - I convince stakeholders about the value of my architectural solution. - Stakeholders approach me to discuss about architectural issues. - I notify stakeholders about actions that deviate from the architecture. - I check whether proposals from stakeholders are in line with the architecture. - I judge whether architectural proposals could continue or be executed.
A4: Documentation - I create documents that describe architectural solutions. - I use (parts of) existing documents while creating new deliverables. - I use templates to store architectural knowledge. - I write vision documents to inform stakeholders. - I write progress reports about the architecture to stakeholders.
A5: Knowledge Acquisition - I learn from my colleagues about architectural principles. - Colleagues learn from me about architectural principles. - I read (scientific/professional) literature on architecture. - I keep my knowledge up-to-date by searching relevant information on Intranet or Internet. - I expand my knowledge by visiting conferences and other events about architecture.

Table 2.1: Architecting Activities

2.2 Support Methods

We also wanted to know the kind of support architects require with respect to sharing architectural knowledge. Effective support for sharing architectural knowledge is crucial, not only to assist individual architects, but also to improve the quality of architectural designs by leveraging all knowledge assets available in the organization.

A large part of the architectural knowledge produced is related to the design decisions, so that explicit support for *management of these decisions* is warranted. Recently, various researchers have proposed tools that help storing concepts like design decisions, rationale, and related architectural knowledge [1, 5, 24] These tools include specialized templates, overviews and storage facilities for architectural knowledge concepts. Such support for management of design decisions could greatly help architects who are involved in decision making. While working on the 'backlog' of open issues and challenges they can easily organize their thought processes and store the architectural knowledge they produce [20].

In addition to assisting architects in producing architectural knowledge, support during the consumption of such knowledge is equally important. The need for a more balanced view on architectural knowledge sharing, in which support for both producing and consuming architectural knowledge is included, has been discussed before [32]. In a recent study it became clear that one of the main requirements architects have is support in retrieving the right architectural knowledge at any time [14]. Probably, support for *efficient searching* of architectural knowledge will greatly boost potential reuse (reusable assets are better accessible) and will stimulate learning among architects (they can find what they need more easily).

Due to the size and complexity of most software systems, it is often infeasible for one architect to be responsible for everything alone. Consequently, the architect-role is often fulfilled by multiple collaborating architects. Liang et al. provide a proper introduction to how a collaborative architecting process would look like [35]. To foster sharing of architectural knowledge between all these architects calls for support methods that focus on *community building*. Architects often do not know what experienced colleagues from other teams work on, what their experience or expertise is, or what their interests are. We argue that support for community building alleviates this situation and increases the amount of architectural knowledge shared. With the advent of 'Web 2.0', this is much easier to implement than in the past. The need for systems that enable such support is explicated in [13]. Likewise, systems such as Wikis can be employed to improve networking by letting stakeholders share ideas, discuss progress, or collaboratively produce architectural knowledge in a variety of formats [16].

Another main type of support in the architecting process relates to *intelligent advice* and support. Architects could benefit from intelligent support during almost all their core activities: a) during production activities such as writing an architectural description, b) directly after producing architectural knowledge (in terms of feedback on what they did), and c) during reviewing and evaluation activities. Intelligent or pro-active support helps architects to leverage and share the architectural knowledge assets available by taking over certain tasks. This might result in a software assistant who thinks together with the practitioners and suggests ideas, challenges decisions, etc. Implementations of pro-active and intelligent assistants are not often seen in practice yet. In a research setting, examples start to emerge. One example is provided by Garlan et al., who designed a personal cognitive assistant called RADAR, which could help architects to accomplish their high-level goals, coordinating the use of multiple applications, automatically handling routine tasks, and, most importantly, adapting to the individual needs of a user over time [17]. Another example is the Knowledge Architect tool suite that, among other things, assists architects in writing documentation, and that can perform quantitative architectural analysis [35].

The last type of support we envision relates to advanced *management of architectural knowledge concepts*. In addition to intelligent support, architects benefit from (semi)-automatic interpretation of stored knowledge to enrich it. Text mining services (e.g., [11]) could be employed to automatically sift through existing architectural knowledge stored (e.g., in a database) looking for new patterns, define best practices, or locate trends. Based on the findings, additional meta-data could be generated by such a service and eventually presented to the architect. Consequently, searches executed could deliver more in-depth results that can then be shared, because part of the interpretation of the 'raw' architectural knowledge has already been done.

Table 2.2 shows the main support methods we distinguish (i.e. decision management, search efficiency, community building, intelligent advice, and knowledge management), indicated by specific requirements that served as items on our Web survey.

S1: Decision Management
- An overview of the most important architectural decisions. - An overview of the relations between taken decisions. - Templates for codification of architectural decisions. - Insight into conflicts between architectural decisions. - An overview of changes through time of certain decisions. - A repository to store architectural decisions.
S2: Search Efficiency
- Search methods for existing architectural guidelines. - Retrieving all documentation related to a specific architectural subject. - Search facilities for decisions within a specific project. - Retrieving relevant information within a project. - Overview of important events related to architecture.
S3: Community Building
- A central system to hold discussions with stakeholders. - Notify colleagues about relevant documentation. - A central environment to collaborate with colleagues on arch. issues. - Retrieve information about the expertise of colleagues. - Acquire insight into which architectural projects colleagues worked on.
S4: Intelligent Advice
- Concrete feedback during the writing process of arch. documentation. - Specific advices on which architectural decisions need to be taken. - Suggestions on how to use architectural guidelines in a specific project. - Being notified about the availability of new architectural publications. - Being sent an overview of the status of an architectural project.
S5: Knowledge Management
- Automatic retrieval of architectural guidelines within projects. - Simple methods to annotate architectural knowledge concepts in documents with meta-data. - A automatically generated overview of open design issues in documents. - A system that tracks overlap among codified architectural guidelines. - Central maintenance of architectural guidelines and best practices.

Table 2.2: Support Methods for SharingArchitectural Knowledge

2.3 Architectural Knowledge Sharing Patterns

Table 2.1 and Table 2.2 show the factors in our theory on architecting activities and support for architectural knowledge sharing. To make it a theory, we hypothesized that specific support might be more effective, or appreciated, during specific architecting activities. This means that certain 'patterns' exist that describe which type of support fits which type of architecting activity best.

We hypothesized that six of such patterns exist, which are depicted in Figure 2.1. For each pattern we have formulated a hypothesis that was tested with our survey.

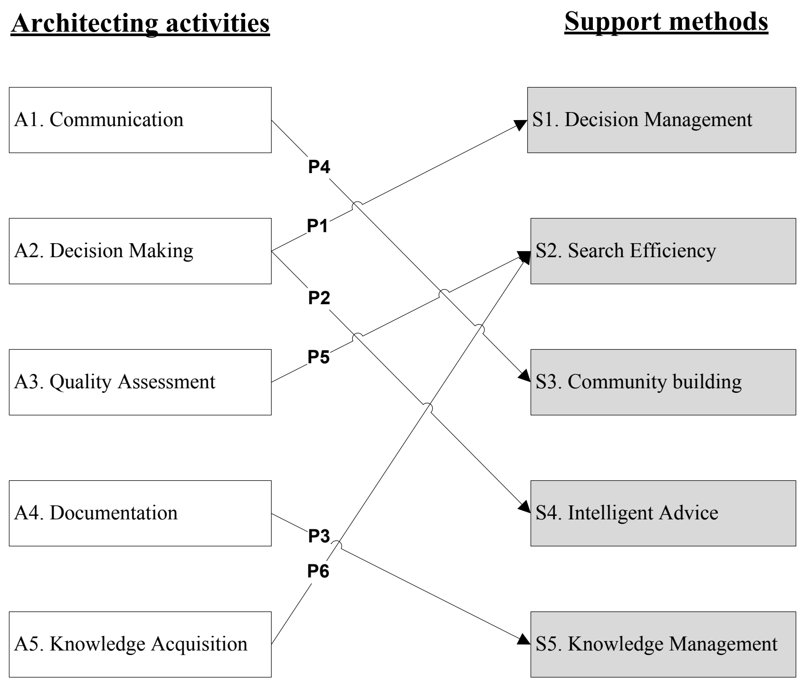

Figure 2.1: Patterns of Architectural Knowledge Sharing Support

P1 Architects involved in taking architectural decisions (A2) benefit particularly from support for management of architectural decisions (S1). We argue that practitioners who are often involved in decision-making could benefit greatly from explicit support for working on a decision 'backlog', by having overviews of open issues, conflicting decisions, traces between requirements and solutions, etc.

P2 Architects involved in taking architectural decisions (A2) benefit particularly from support for intelligent advice (S4). We argue that during the thought processes and tradeoffs inherent to decision making, pro-active or just-in-time support might benefit architects greatly. This could include (automated) advice on which decisions to take in a specific situation, or tips about how to apply architectural patterns, styles or tactics.

P3 Architects involved in documenting architectural knowledge (A4) benefit particularly from support for maintenance and overview of architectural knowledge (S5). Architects who are often involved in codifying knowledge in artifacts such as architectural descriptions are directly helped by templates, models, frameworks etc. Also easy access to other codification methods such as central repositories for guidelines and repositories lower the threshold of producing something and eases the storing process.

P4 Architects involved in communication with colleagues or other stakeholders (A1) benefit particularly from support for community building (S3). We expect architects who are frequently involved in communication with colleagues or other stakeholders to be more enthusiastic about community building, sharing knowledge, and networking using for example social networks (e.g., wikis, blogs). After all, methods such as people directories, yellow pages, but also discussion facilities are more interesting for people that often use them and also lead to greater benefits than when there is less collaboration with others.

P5 Architects involved in quality assessments (A3) benefit particularly from support for efficient retrieval of (or searching for) architectural knowledge (S2). During reviews and assessments quickly retrieving information about the status of a project, the set of principles applied, or the key architectural decisions taken greatly benefit architects. This would also lead to higher-quality results during the evaluation, because it is based on more accurate and complete architectural knowledge.

P6 Architects involved in expanding their knowledge on architecture (A5) benefit particularly from support for efficient retrieval architectural knowledge (S2). Architects could learn more effectively (i.e. faster, more to-the-point) if they have the means to quickly retrieve the architectural knowledge they need at a specific point in time. Just-in-time architectural knowledge allows them to quickly find new domain knowledge, get the status of a project, or read through the main deliverables of a project. To put it differently, by having good support for retrieving such knowledge, the practitioner can enrich and refine the knowledge himself by combining inputs from different sources, so that the knowledge internalization process is significantly improved.

Chapter 3

Research Methodology

In total 279 Dutch architects from four IT organizations were included in this research. These organizations include:

1. CAP, a large international software development organization that operates in the public, finance and industry domain.

2. CON, an international consultancy firm specialized in infrastructure and maintenance of software systems.

3. CSM, an international IT Service Organization for consulting, system integration and managed operations.

4. CGO, a central Dutch governmental organization for the governance of ICT architectures in the public domain.

Our study is conducted according to a structured design, which is elaborated upon in the remainder of this chapter. Our research methodology includes all important elements required to properly design, administer and analyze a survey [26]. Our survey consists of a pilot study, followed by the main study, both of which are further explained in the following sections.

3.1 Pilot Study

The theoretical framework presented in Chapter 2 is not something we built up overnight. An earlier theory consisting of six categories of architecting activities and six different support methods was presented in [15]. The main goal of the pilot study was to see whether this earlier theory was sufficiently strong for usage in the main study, and to improve – or refactor – it where necessary. To this end, we made constructs for all activities and support methods. Inspired by relevant literature and prior case studies (see [15] for more details) we formulated 10 to 15 questions (i.e. 'items') for each of these constructs. After iteratively scrutinizing these items on clarity and redundancy, some were removed, leaving each construct between 5 and 10 items.

Because 10 items per construct would make the questionnaire of the main study too long, we decided to let a small percentage of the total sample (N=8) pretest all items in a pilot questionnaire. The participating architects were excluded from the main study. To test the internal consistency of the items we performed reliability analysis on the item scores of the pilot questionnaire, and selected the ones with the strongest internal consistency (i.e. highest Cronbach's alpha). To make sure that items did not indicate other activities or support methods as well, we performed factor analysis. This way we could sharpen our concepts.

With respect to the theory on architecting activities, we updated our theoretical framework of architecting activities slightly. It turned out during factor analysis that no obvious distinction could be made between communication within a team (with colleagues) and communication outside the team (with other stakeholders). The former category showed much stronger in the factor analysis so we decided to cancel the latter one. So this 'external' communication is not one of the most striking activities acknowledged by practicing architects. We thus labeled this category as 'Communication'.

Also some of the subactivities turned out to be 'less explaining' for the factors/scales identified. The factor analysis showed us which subactivities (i.e. items constructed around these subactivities) had a better internal consistency (Cronbach's alpha) and scored better on the underlying factor. Based on these insights we were able to select the best explaining items/subactivities for each of the categories.

Some items scored high on a factor, but not on the factor they were initially designed for. For example the item 'I convince stakeholders about the value of my architectural solution.' initially belonged to communication with stakeholders outside the team', but the factor analysis indicated that it strongly correlated with items related to 'controlling quality' (now relabeled as Quality Assessment). In this case the relation is not hard to see. Stressing architectural solutions to stakeholders is usually done with the goal of ensuring the adherence to and thus the quality of the architecture. In this case, we added the item to the new category. In some other cases, although the factor analysis indicated relations between two items, there was no intuitive mapping possible. In this case, we removed that item from the final survey because it would contaminate the results.

Finally, some categories were relabeled because by having removed some items these categories could be defined in a more compact way such as 'Documentation' and 'Knowledge acquisition'. The improved, i.e. refactored, framework of activities thus contained five categories as depicted in (Table 2.1). The subactivities are phrased as the actual statements as put in the final survey that was used in the main study.

Based on a pilot survey, we also updated our theoretical framework of support for architectural knowledge sharing slightly. Just as in the improvement of our model of architecting activities we looked at contaminating items, factors emerged based on the pilot survey, internal consistency, and factors that turned out to be unconfirmed. It turned out that no separate factors for support for codification or enrichment of architectural knowledge could be identified based on the pilot study results. Furthermore, the category about codification of architectural knowledge was very contaminated. Examining the items led us to the insight that apparently the sub-methods describing the codification process very much linked to either creating solutions (i.e. 'managing decisions') or automated support for storing knowledge ('intelligent support'). We decided to remove this category all together.

The scale ‘Enrichment of knowledge’ was the strongest one in the factor analysis. However, the items with the highest internal consistency were mostly focused on acquiring overviews of architectural knowledge or on maintaining knowledge concepts (such as guidelines) in repositories. We therefore relabeled this category into ‘Knowledge Management’. Two other categories were relabeled for similar reasons (i.e. based on the items that were selected). We relabeled searching architectural knowledge’ to ‘Search Efficiency’ and ‘intelligent support’ to ‘Intelligent Advice’. So, coincidentally, just as with the theory on activities, we had to change our theory by going from six to five constructs. For each of these support method constructs the five or six most explanatory items were selected, as depicted in Table 2.2.

3.2 Main Study

To answer our main research question we first validated our theory on architecting activities and support methods for architectural knowledge sharing. This was a necessary prerequisite before we could validate the earlier defined patterns for architectural knowledge sharing support. To obtain a better understanding of which support methods architects prefer, we assessed how they would prioritize these methods. The questionnaire of our main study therefore consisted of three parts:

1. **Part 1: Architecting activities.** Each of the five architecting activities was treated as a separate scale, consisting of the (four to six) items as depicted in Table 2.1. Each item described a sub-activity (e.g., *‘I inform colleagues about the results of my work’*) and respondents replied to what extent that activity related to their daily work by scoring a 6-point Likert-type rating scale (1 = totally disagree, 6 = totally agree).

2. **Part 2: Support methods.** Each of the five support methods for sharing architectural knowledge was treated as a separate scale, consisting of the (five to six) items representing sub-methods, as depicted in Table 2.2. Respondents indicated on 6-point Likert-type rating scales to what degree these support methods were helpful in their daily work. A sample item was *‘In my daily work, I benefit from an overview of the most important architectural decisions’*

3. **Part 3: Prioritization of support methods.** The respondents prioritized the five types of support methods in order of importance for *one particular* activity alone. They ranked the five methods between 1 and 5, where a score of ‘1’ corresponded to the highest ranked method and ‘5’ corresponded to the lowest ranked method for that architecting activity. We made sure that no score could be given twice and that no missing values were allowed.

If we would have opted for following a ‘classical’ survey approach we could have translated our theory on architecting activities and support methods in a number of ‘blocks of text’, pose these to the respondents and ask them to prioritize the methods for each activity (this is Part 3). Our research is more advanced in that we use Part 1 and Part 2 to first pose questions that validate our theory on activities and support methods. Only then we can interpret the results of Part 3 in the right way. For example if Part 1 shows that architects do not deem activity 4 relevant (i.e. they give low scores to all sub-activities in that scale), then the prioritization exercise for activity 4 is no longer relevant either. So although Part 3 is the most important part of our study, Part 1 and Part 2 are important prerequisites to validate our theory.

In addition to the questions of the three parts, our (anonymous) questionnaire contained some additional questions including the respondent's role (which 'type' of architect), and his or her experience (number of work years). This allowed us to analyze the impact of these additional factors on respectively the architecting activities they are involved in (Part 1), the support methods they like (Part 2) and their prioritization of those support methods (Part 3).

To prevent bias and fatigue effects we decided to give each architect only one prioritization question in Part 3 of the survey [21]. To prevent bias we also randomized the order of the question screens of Part 1 and Part 2 as much as possible. We have constructed the following experimental design to accommodate for these requirements:

- We split up the sample in 5 equally large homogeneous groups of architects. Each of these groups got a different questionnaire. Of these 5 questionnaires Part 1 and Part 2 were identical, except for the ordering of the screens. The difference was in Part 3: each questionnaire only contained *'one'* prioritization question, i.e. selected one architecting activity and asked the architects to prioritize the support methods for this activity. This prevented too much repetition (and possible contaminated results), which would have occurred if the same respondent had to rank the same methods five times, once for each activity [21].

- To verify whether the ordering of the screens in Part 1 and Part 2 did not lead to a bias in how the support methods in Part 3 were ranked, we made two versions for each of the 5 groups mentioned above. Each of these versions had a distinct ordering of the question screens, as is depicted in Table 3.1. So in the end the respondents were divided equally over 10 questionnaires (Q1-Q10), but as soon as the test for bias has been negative we can form five bigger groups again (by combining Q1+Q6, Q2+Q7, etc.).

- Finally, all questions in the screens of Part 1 and Part 2 were randomized, which means that they were posed to each respondent in a different order.

- All respondents start from the same introduction screen in the web survey tool Examine [1], but as soon as they start with the actual questionnaire, the 'experimental design' feature of Examine redirects the respondent to one of the ten versions (Q1-Q10). The advantage of this approach is that there is only one link to our (anonymous) questionnaire, which was thus easy to place in the invitation emails send out in the four participating organizations.

[1] http://examine.vu.nl/

10 versions (Q1,..,Q10). Each questionnaire has only one prioritization question in Part 3

	Part 1: Architecting activities					Part 2: Support for AK sharing					Part 3: Prioritize support	
Quest.	Act1	Act2	Act3	Act4	Act5	Sup1	Sup2	Sup3	Sup4	Sup5	Ranking question	
Q1	1	2	3	4	5	1	2	3	4	5	**Act1**	
Q2	1	2	3	4	5	1	2	3	4	5	**Act2**	
Q3	1	2	3	4	5	1	2	3	4	5	**Act3**	
Q4	1	2	3	4	5	1	2	3	4	5	**Act4**	
Q5	1	2	3	4	5	1	2	3	4	5	**Act5**	
Q6	5	4	3	2	1	5	4	3	2	1	**Act1**	
Q7	5	4	3	2	1	5	4	3	2	1	**Act2**	
Q8	5	4	3	2	1	5	4	3	2	1	**Act3**	
Q9	5	4	3	2	1	5	4	3	2	1	**Act4**	
Q10	5	4	3	2	1	5	4	3	2	1	**Act5**	

N.B. within each category the questions are shown in random order

Table 3.1: Experimental Design: Different versions of the same questionnaire

3.2.1 Measurements

In our main study we will test a number of hypotheses. The most important ones refer to the architectural knowledge sharing patterns identified earlier. We will test P1 to P6 by looking at whether there is a significant causality between architects scoring high on certain questions about architecting activities on the one hand, and their favorability on specific support methods on the other hand. This will answer research question RQ3.

Factor analysis allows us to see whether there is evidence that these activities and support methods are sufficiently distinct from each other, or that less or more factors can be identified. By closely examining the factors we will determine to what extent our aggregated theory of activities holds.

Apart from aforementioned hypotheses, we examine several other interesting aspects:

- whether the type of organization an architect works for has an effect on which activities he undertakes.
- whether the type of organization an architect works for has an effect on which support methods he deems valuable.
- whether the type of organization an architect works for has an effect on which support category he ranks highest.
- whether work experience has an effect on which activities architects do.
- whether work experience has an effect on which support architects deem valuable.
- whether work experience has an effect on which support category the architects ranked highest.
- whether the type of architect has an effect on which activities architects do.
- whether the type of architect has an effect on which support architects deem valuable.
- whether the type of architect has an effect on which support category the architects ranked highest.

Chapter 4

Data Analysis

Our main study was conducted in February 2009 at the four IT organizations in parallel. In total 271 architects from these organizations were included in the main study, 156 of whom responded (see Table 4.1).

Response rate

Organization	Sample	Responses	Response rate
CAP	56	42	75,0%
CON	39	20	51,3%
CSM	171	76	44,4%
CGO	5	4	80%
Total	**271**	**142**	**52,4%**

Table 4.1: Response rate

The response rates have been calculated after we obtained more detailed knowledge about the participating architects. Although we spent quite some effort to distinguish within the participating organizations who would qualify for being 'an architect' according to our definition, we still received some replies from respondents that they are not practicing architect (anymore), left the company, etc. Based on this information we updated our list of respondents and response rate accordingly.

The architects from CSM were invited through our contact person, so we never saw the list they used. He informed us that the list was compiled out of attendees from previous architecture events. From CAP and CON we know that although this list is a good start, still several project managers, account managers or other people may be part of it, but in CSM we were not able to filter them out. The 171 participating architects is therefore probably too large, which might explain the lower response rate we got from this organization.

From the 156 respondents, 13 were discarded in the analysis because their records were completely filled with missing values. One respondent was removed from the analysis because s/he produced scores of 1 alone. This left us with 142 respondents for the scale and factor analysis, which accounts for a response rate of 52.4%. Remaining missings were treated as subject-missing.

Because our survey targeted 'the architecture communities' of the four organizations, various 'types' of architects were included. From the 142 respondents, 48 labeled themselves as Enterprise architects, 63 as IT or software architect, and the remaining ones (31) were either information analysts, infrastructure architects, or application designers. Their average relevant working experience was 6 years; 26 architects indicated less than 3 years of architecting experience, 89 architects 5 years or more.

4.1 Scale analysis

To assess whether our data was internally consistent and that the items indicated our constructs of activities and support methods, we started with conducting scale analysis. To this end, we assessed the psychometric quality of the 5 Activity scales (Communication, Decision Making, Quality Assessment, Documentation, Knowledge acquisition) and the 5 Support scales (Decision Management, Search Efficiency, Community Building, Intelligent Advice, Knowledge Management). Scales had between 4 and 6 items each. We tested whether items correlated sufficiently with their own scale by means of Corrected Item-Total Correlations and regular Cronbachs alpha (indicating reliability).

Scale	*Cronbach's α*	*# items*	*n*
Communication	.80	4	132
Decision making	.78	4	131
Quality assessment	.83	6	128
Documentation	.76	5	131
Knowledge acquisition	.70	5	136
Decision management	.81	6	123
Search efficiency	.82	5	123
Community building	.82	5	123
Intelligent advice	.81	5	123
Knowledge management	.90	6	122

Table 4.2: Reliability analysis before factor analysis

Item selection was a trade-off among several criteria. We wanted to establish as many items on a scale as possible with a minimum of 2, provided that Cronbachs alpha for a scale was $\geq .60$ and Corrected Item-Total Correlations $\geq .20$. In Table 4.2 the reliability analysis before factor analysis is depicted. In addition, the degree to which items did not correlate with other scales was tested with factor analysis, one time for the Activities scales and one time for the Support scales (PCA, 25 iterations Varimax rotation with Kaiser Normalization, number of factors to be extracted: 5). The rotated component matrix showed that all five factors could be retrieved in the data, except for a limited number of items. We removed 8 items that correlated more strongly with another scale than with their own scale. Three more items were removed that correlated above .40 with other scales irrespective of high correlations with their own scale. In Table 4.3 the resulting set of items is depicted.

The thus revised scales were submitted to reliability analysis again. Scale length was reduced to 3, 4, or 5 items. Regular Cronbach's alphas were between .72 and .90, which is reasonable to good. The scale Knowledge Acquisition had

A1: Communication - I inform colleagues about the results of my work. - My colleagues keep me up-to-date on the results of their work. - I explain existing architectural principles to colleagues. - During discussions I share my knowledge to colleagues.	**S1: Decision Management** - An overview of the most important architectural decisions. - An overview of the relations between taken decisions. - Templates for codification of architectural decisions. - Insight into conflicts between architectural decisions. - An overview of changes through time of certain decisions.
A2: Decision Making - Before I take a decision I weigh the pros and cons of possible solutions. - I think about what impact my decisions have on the current architecture. - I study the reasoning behind taken design decisions.	**S2: Search Efficiency** - Search methods for existing architectural guidelines. - Retrieving all documentation related to a specific architectural subject. - Search facilities for decisions within a specific project. - Retrieving relevant information within a project.
A3: Quality Assessment - I convince stakeholders about the value of my architectural solution. - Stakeholders approach me to discuss about architectural issues. - I notify stakeholders about actions that deviate from the architecture. - I check whether proposals from stakeholders are in line with the architecture. - I judge whether architectural proposals could continue or be executed.	**S3: Community Building** - A central system to hold discussions with stakeholders. - Notify colleagues about relevant documentation. - A central environment to collaborate with colleagues on arch. issues. - Retrieve information about the expertise of colleagues.
A4: Documentation - I use (parts of) existing documents while creating new deliverables. - I use templates to store architectural knowledge. - I write vision documents to inform stakeholders. - I write progress reports about the architecture to stakeholders.	**S4: Intelligent Advice** - Specific advices on which architectural decisions need to be taken. - Suggestions on how to use architectural guidelines in a specific project. - Being sent an overview of the status of an architectural project.
A5: Knowledge Acquisition - I read (scientific/professional) literature on architecture. - I keep my knowledge up-to-date by searching relevant information on Intranet or Internet. - I expand my knowledge by visiting conferences and other events about architecture.	**S5: Knowledge Management** - Automatic retrieval of architectural guidelines within projects. - Simple methods to annotate architectural knowledge concepts in documents with meta-data. - A automatically generated overview of open design issues in documents. - A system that tracks overlap among codified architectural guidelines.

Table 4.3: Selected Items after Reliability Analysis

an alpha of .61, which is suspect but still acceptable. In Table 4.4 the reliability analysis with shortened scales after factor analysis is depicted.

Scale	*Cronbach's α*	*# items*	*n*
Communication	.80	4	132
Decision making	.85	3	131
Quality assessment	.80	5	128
Documentation	.72	4	131
Knowledge acquisition	.61	3	136
Decision management	.80	5	123
Search efficiency	.83	4	123
Community building	.78	4	123
Intelligent advice	.74	3	123
Knowledge management	.90	5	122

Table 4.4: Reliability analysis with shortened scales after factor analysis

4.1.1 Computation of Scale Means and Tests on Control Variables

We calculated the mean agreement scores to items for each shortened scale. To control for the effects of the order of item blocks in the survey, respondents worked with either of two versions with different item-block orders. We tested the effects of Block Order (2) by running a GLM-Repeated Measures for the 5-leveled Activities factor (Communication, Decision Making, Quality Assessment, Documentation, Knowledge Acquisition) on the mean agreement. Yet, as depicted in Table 4.5 the main effect of Block Order was insignificant ($F_{(4,124)} = 1.3$, $p >> .05$) and no significant interaction occurred between Activities and Block Order (Pillai's Trace = .04, $F_{(4,121)} = 1.4$, $p >> .05$). We also tested the effects of Organization (4) and Activities on the mean agreement, yielding a marginally significant main effect of Organization ($F_{(3,122)} = 2.6$, $p = .053$) but with a very small effect size ($\eta_p^2 = .06$). The interaction between Organization and Activities was not significant ($F < 1$) (cf. Table 4.6). We repeated this analysis for Function (10) but this rendered insignificant results for the main effect of Function ($F_{(9,116)} = 1.5$, $p >> .05$) and for the interaction of Function by Support (Pillai's Trace = .40, $F_{(36,464)} = 1.4$, $p > .05$) (cf. Table 4.7).

Tests of Between-Subjects Effects

Measure: Mean_Agreement
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	13215,731	1	13215,731	8559,898	,000	,986
Blockorder	1,958	1	1,958	1,268	,262	,010
Error	191,445	124	1,544			

Multivariate Tests(b)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Activities * Blockorder	Pillai's Trace	,043	1,351(a)	4,000	121,000	,255	,043
	Wilks' Lambda	,957	1,351(a)	4,000	121,000	,255	,043
	Hotelling's Trace	,045	1,351(a)	4,000	121,000	,255	,043
	Roy's Largest Root	,045	1,351(a)	4,000	121,000	,255	,043

a Exact statistic
b Design: Intercept+Blockorder
Within Subjects Design: Activities

Table 4.5: Effects of Block Order on Activities

Tests of Between-Subjects Effects

Measure: Mean_Agreement
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	5101,635	1	5101,635	3426,912	,000	,966
Organization	11,782	3	3,927	2,638	,053	,061
Error	181,621	122	1,489			

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Activities * Organization	Pillai's Trace	,064	,660	12,000	363,000	,790	,021
	Wilks' Lambda	,937	,652	12,000	315,136	,797	,021
	Hotelling's Trace	,066	,645	12,000	353,000	,803	,021
	Roy's Largest Root	,034	1,029(b)	4,000	121,000	,395	,033

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+Organization
Within Subjects Design: Activities

Table 4.6: Effects of Organization on Activities

We conducted the same analyses for the 5-leveled Support factor (Decision Management, Search Efficiency, Community Building, Intelligent Advice, and Knowledge Management). Again, the main effect of Block Order was insignificant ($F_{(1,120)} = 1.5$, $p >> .05$) and no significant interaction occurred between Support and Block Order (Pillai's Trace = .06, $F_{(4,117)} = 2.0$, $p >> .05$) (cf. Table 4.8). The main effect of Organization (4) was not significant either ($F < 1$) nor was the interaction between Organization and Support (Pillai's Trace = .16, $F_{(12,351)} = 1.6$, $p > .05$) (cf. Table 4.9). The main effect of Function (10) also remained insignificant ($F_{(9,112)} = 1.1$, $p >> .05$) as did the interaction of Function by Support (Pillai's Trace = .23, $F_{(36,448)} = .76$, $p >> .05$) (cf. Table 4.10).

Tests of Between-Subjects Effects

Measure: Mean_Agreement
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	8498,137	1	8498,137	5679,316	,000	,980
Function	19,828	9	2,203	1,472	,166	,103
Error	173,574	116	1,496			

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Activities * Function	Pillai's Trace	,395	1,411	36,000	464,000	,061	,099
	Wilks' Lambda	,649	1,444	36,000	425,201	,050	,102
	Hotelling's Trace	,476	1,474	36,000	446,000	,041	,106
	Roy's Largest Root	,294	3,785(b)	9,000	116,000	,000	,227

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+Function
Within Subjects Design: Activities

Table 4.7: Effects of Function on Activities

In further analyses, therefore, data will be pooled over Block Order, Organization, and Function.

Tests of Between-Subjects Effects

Measure: Mean_Agreement
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	13008,886	1	13008,886	9522,355	,000	,988
Blockorder	2,074	1	2,074	1,518	,220	,012
Error	163,937	120	1,366			

Multivariate Tests(b)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Support * Blockorder	Pillai's Trace	,064	1,993(a)	4,000	117,000	,100	,064
	Wilks' Lambda	,936	1,993(a)	4,000	117,000	,100	,064
	Hotelling's Trace	,068	1,993(a)	4,000	117,000	,100	,064
	Roy's Largest Root	,068	1,993(a)	4,000	117,000	,100	,064

a Exact statistic
b Design: Intercept+Blockorder
Within Subjects Design: Support

Table 4.8: Effects of Block Order on Support

Tests of Between-Subjects Effects

Measure: Mean_Agreement
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	4940,564	1	4940,564	3571,225	,000	,968
Organization	2,765	3	,922	,666	,574	,017
Error	163,246	118	1,383			

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Support * Organization	Pillai's Trace	,158	1,623	12,000	351,000	,083	,053
	Wilks' Lambda	,848	1,629	12,000	304,553	,083	,053
	Hotelling's Trace	,172	1,627	12,000	341,000	,082	,054
	Roy's Largest Root	,108	3,161(b)	4,000	117,000	,017	,098

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+Organization
Within Subjects Design: Support

Table 4.9: Effects of Organization on Support

Tests of Between-Subjects Effects

Measure: Mean_Agreement
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	8409,676	1	8409,676	6178,069	,000
Function	13,555	9	1,506	1,106	,364
Error	152,456	112	1,361		

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.
Support * Function	Pillai's Trace	,230	,760	36,000	448,000	,843
	Wilks' Lambda	,787	,751	36,000	410,211	,853
	Hotelling's Trace	,249	,742	36,000	430,000	,863
	Roy's Largest Root	,114	1,419(b)	9,000	112,000	,188

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+Function
Within Subjects Design: Support

Table 4.10: Effects of Function on Support

4.2 What Architects Do

We are now ready for answering research question RQ1, which was phrased as follows:

- *RQ1: With which activity do architects agree the most as an illustration of their work? In other words, what is their core activity?*

To answer this question, we ran a GLM-Repeated Measures for the 5-leveled Activities factor (Communication, Decision Making, Quality Assessment, Documentation, Knowledge Acquisition) on the mean agreement scores with Work Years as covariate.

According to the multivariate tests, the main effect of Activities was significant with a considerable effect size (Pillai's Trace = .51, $F_{(4,121)} = 31.4$, $p = .000$, $\eta_p^2 = .51$). Table 4.11 shows that Decision Making received the highest agreement scores (M = 5.21, SD = .65) and Documentation the lowest (M = 3.98, SD = .89). Paired-samples t-tests (cf. Table 4.12) showed that all differences among the Activities were significant ($t > 2.7$, $p < .009$), except for the contrast between Communication and Knowledge Acquisition ($t_{125} = -.38$, $p >> .05$).

	Mean	Std. Deviation	Correlation with Work Years	N
Work years (experience in architecture projects)	6.22	4.110		142
Communication	4.6155	.67423	.20*	132
Decision Making	5.2087	.65285	.18*	131
Quality Assessment	4.4484	.82204	.37**	128
Documentation	3.9847	.89052	.23**	131
Knowledge Acquisition	4.6152	.74135	.15	136
Decision Management	4.8374	.58188	.15	123
Search Efficiency	4.8943	.60902	.09	123
Community Building	4.5325	.74586	.10	123
Intelligent Advice	4.5230	.64030	.10	123
Knowledge Management	4.3033	.90077	.15	122

Table 4.11: Descriptives and Correlations of Activities and Support

However, these effects were modulated by the number of Work Years that respondents worked on architectural projects (M = 6.22 years, SD = 4.11). The interaction between Activities and Work Years was significant albeit with small effect size (Pillai's Trace = .09, $F_{(4,121)} = 2.9$, $p = .026$, $\eta_p^2 = .09$) (cf. Table 4.13). Further inquiry with Pearson correlations into the relationship between this covariate and the different Activities showed that particularly agreement to Quality Assessment and Documentation increased significantly with the number of Work Years. This was also the case for Communication and Decision Making but not for Knowledge Acquisition (see Table 4.11).

Paired Samples Test

		Paired Differences						df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Upper	Lower	t	Std. Deviation	Std. Error Mean
Pair 1	Communication - Decision making	-,57697	,69971	,06113	-,69792	-,45603	-9,438	130	,000
Pair 2	Communication - Quality assessment	,17852	,74595	,06593	,04805	,30899	2,708	127	,008
Pair 3	Communication - Documentation	,63294	,86121	,07672	,48109	,78478	8,250	125	,000
Pair 4	Communication - Knowledge acquisition	-,02712	,80158	,07141	-,16845	,11421	-,380	125	,705
Pair 5	Decision making - Quality assessment	,76250	,71801	,06346	,63692	,88808	12,015	127	,000
Pair 6	Decision making - Documentation	1,21825	,85087	,07580	1,06823	1,36828	16,072	125	,000
Pair 7	Decision making - Knowledge acquisition	,55820	,75875	,06759	,42442	,69198	8,258	125	,000
Pair 8	Quality assessment - Documentation	,44841	,83916	,07476	,30046	,59637	5,998	125	,000
Pair 9	Quality assessment - Knowledge acquisition	-,21164	,85138	,07585	-,36175	-,06153	-2,790	125	,006
Pair 10	Documentation - Knowledge acquisition	-,65649	,82819	,07236	-,79964	-,51333	-9,073	130	,000

Table 4.12: Paired Samples Test of Activities

Our survey results thus indicate that interestingly enough architects spend most of their time on making architectural decisions and least on documenting the results. Apparently architects play an important role in the architecting process but for various reasons (e.g., available time, lack of interest) neglect documenting what they have decided. This has the risk of leading to substantial architectural knowledge vaporization [3] and substantially decreases the chance for effectively reusing architectural knowledge.

When looking at the effect of Work Years on the activities, our results showed that particularly agreement to Quality Assessment and Documentation increased significantly with the number of Work Years. This indicates that more experienced architects seem more often involved in auditing activities, and – maybe related to this – spend more energy in documenting their results. Maybe the advantages of retrieving or reusing codified knowledge has proven itself to these experienced architects over the years.

Multivariate Tests(b)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Activities	Pillai's Trace	,509	31,391(a)	4,000	121,000	,000	,509
	Wilks' Lambda	,491	31,391(a)	4,000	121,000	,000	,509
	Hotelling's Trace	1,038	31,391(a)	4,000	121,000	,000	,509
	Roy's Largest Root	1,038	31,391(a)	4,000	121,000	,000	,509
Activities * WorkYears	Pillai's Trace	,086	2,863(a)	4,000	121,000	,026	,086
	Wilks' Lambda	,914	2,863(a)	4,000	121,000	,026	,086
	Hotelling's Trace	,095	2,863(a)	4,000	121,000	,026	,086
	Roy's Largest Root	,095	2,863(a)	4,000	121,000	,026	,086

a Exact statistic
b Design: Intercept+WorkYears
Within Subjects Design: Activities

Table 4.13: Interaction between Activities and Work Years

4.3 What Architects Need

The next step is looking at the support methods in order to answer research question RQ2, which was phrased as follows:

- *RQ2: With which kind of support do architects agree the most as a contribution to their work? In other words, what is their core need?*

To examine what kind of support methods architects desire, we studied the data of Part 2 of our questionnaire. We ran a GLM-Repeated Measures for the 5-leveled Support factor (Decision Management, Search Efficiency, Community Building, Intelligent Advice, and Knowledge Management) on the mean agreement scores with Work Years as covariate. The main effect of Support was significant with medium to low effect size (Pillai's Trace = .21, $F_{(4,117)} = 7.7$, $p = .000$, $\eta_p^2 = .21$). Table 4.11 shows that Search Efficiency (M = 4.89, SD = .61) raised the highest agreement scores and Knowledge Management the lowest (M = 4.30, SD = .90). The interaction between Support and Work Years was not significant ($F < 1$) (see Table 4.14).

Multivariate Tests(b)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Support	Pillai's Trace	,209	7,730(a)	4,000	117,000	,000	,209
	Wilks' Lambda	,791	7,730(a)	4,000	117,000	,000	,209
	Hotelling's Trace	,264	7,730(a)	4,000	117,000	,000	,209
	Roy's Largest Root	,264	7,730(a)	4,000	117,000	,000	,209
Support * WorkYears	Pillai's Trace	,008	,229(a)	4,000	117,000	,922	,008
	Wilks' Lambda	,992	,229(a)	4,000	117,000	,922	,008
	Hotelling's Trace	,008	,229(a)	4,000	117,000	,922	,008
	Roy's Largest Root	,008	,229(a)	4,000	117,000	,922	,008

a Exact statistic
b Design: Intercept+WorkYears
Within Subjects Design: Support

Table 4.14: Interaction between Support and Work Years

To further explore the main effect of Support, paired-samples t-tests showed that all differences among the kinds of Support were significant ($t > 2.7$, $p < .008$), except for the contrast between Decision Management and Search Efficiency ($t_{122} = -.38$, $p >> .05$) and between Community Building and Intelligent Advice ($t_{122} = .15$, $p >> .05$) (see Table 4.15). This means that Search

Efficiency and Decision Management were equally agreed upon as the kind of support that is most needed, irrespective of Work Years.

Paired Samples Test

		Paired Differences						df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Upper	Lower	t	Std. Deviation	Std. Error Mean
Pair 1	Decision management - Search efficiency	-,05691	,60792	,05481	-,16542	,05160	-1,038	122	,301
Pair 2	Decision management - Community building	,30488	,72189	,06509	,17602	,43373	4,684	122	,000
Pair 3	Decision management - Intelligent advice	,31436	,70053	,06317	,18932	,43940	4,977	122	,000
Pair 4	Decision management - Knowledge management	,53443	,81330	,07363	,38865	,68020	7,258	121	,000
Pair 5	Search efficiency - Community building	,36179	,67006	,06042	,24219	,48139	5,988	122	,000
Pair 6	Search efficiency - Intelligent advice	,37127	,60736	,05476	,26286	,47968	6,780	122	,000
Pair 7	Search efficiency - Knowledge management	,59016	,91106	,08248	,42687	,75346	7,155	121	,000
Pair 8	Community building - Intelligent advice	,00949	,68795	,06203	-,11331	,13228	,153	122	,879
Pair 9	Community building - Knowledge management	,23156	,83342	,07545	,08218	,38094	3,069	121	,003
Pair 10	Intelligent advice - Knowledge management	,21585	,87252	,07899	,05946	,37224	2,732	121	,007

Table 4.15: Paired Samples Test of Support

We find it rather contradictory that although architects do not document much, these results indicate that they do wish for efficient support to retrieve useful – previously stored – knowledge. It seems architects really wish to use codified knowledge, but contribute little to codify it in the first place. A proper balance between producing and consuming architectural knowledge, as deemed important in [32], is lacking.

Another surprise related to architects' opinion about the various support methods as indicated by the low score for Knowledge Management. Considering that architects make a lot of decisions but are not that eager to properly document them, we expected that they would be more enthusiastic about automated support for these activities. One would say that easy annotation of architectural knowledge in documents would make the codification tasks a little less cumbersome. Yet, architects do not fancy such pro-active, or automated support. Perhaps they wish to remain in the driver's seat, and do not trust a system that manipulates and processes architectural knowledge independently. This is in line with an earlier study that indicated that one of the desired properties for architectural knowledge sharing support is that it is descriptive in nature and not prescriptive [13].

4.4 Preferences in Support Methods

We examined which support methods architects prefer (RQ3) by testing the patterns hypothesized in Figure 2.1 by retrieving them in the data of Part 1 and Part 2 of our questionnaire.

Based on the six patterns defined in Section 2.3 we can formulate six corresponding hypotheses. We formulated six regression models in which the level of agreement to activities would predict the level of agreement to the kind of support needed: $A1 \rightarrow S3$, $A2 \rightarrow S1$, $A2 \rightarrow S4$, $A3 \rightarrow S2$, $A4 \rightarrow S5$, $A3\&A5 \rightarrow S2$. These hypothesized causalities relate to the arrows drawn in Figure 2.1. Then we performed linear regression analysis (method Enter, entry: .05, removal: .10) to evaluate these assumptions.

In the following six subsections, we test these hypotheses in turn.

4.4.1 Hypothesis P1

Relation to test: $A2 \rightarrow S1$. Decision Making was the predictor for Decision Management, who served as the dependent. Decision Making accounted for a significant quantity of the variability in agreement to Decision Management ($R^2 = .063$, $R^2_{adj} = .056$, $F_{(1,121)} = 8.2$, $p = .005$). As depicted in Table 4.16, Decision Making explained 25% of the variance in Decision Management (standardized $\beta = .25, t = 2.9, p = .005$). Therefore, this hypothesis is accepted.

This means that architects who make decisions are particularly in favor of decision management. This is in line with our assumption that backlog management and effective management of design decisions makes it easier for architects to have a proper overview of the solution space, reusable assets, potential conflicts, stakeholder demands, etc.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,252(a)	,063	,056	,56548

a Predictors: (Constant), Decision making

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2,616	1	2,616	8,181	,005(a)
	Residual	38,692	121	,320		
	Total	41,308	122			

a Predictors: (Constant), Decision making
b Dependent Variable: Decision management

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3,686	,406		9,082	,000
	Decision making	,221	,077	,252	2,860	,005

a Dependent Variable: Decision management

Table 4.16: Testing Hypothesis P1

4.4.2 Hypothesis P2

Relation to test: $A2 \rightarrow S4$. Decision Making accounted for a significant quantity of the variability in agreement to Intelligent Advice ($R^2 = .035, R^2_{adj} = .027, F_{(1,121)} = 4.3, p = .039$). As depicted in Table 4.17, Decision Making explained 19% of the variance in Intelligent Advice (standardized $\beta = .19, t = 2.1, p = .039$). Therefore, this hypothesis is accepted.

This means that intelligent advice such as specific advice on which design decision to take, or suggestions about which architectural guidelines or styles to use, is especially helpful for architects who are busy taking design decisions.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,186(a)	,035	,027	,63169

a Predictors: (Constant), Decision making

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1,735	1	1,735	4,348	,039(a)
	Residual	48,283	121	,399		
	Total	50,018	122			

a Predictors: (Constant), Decision making
b Dependent Variable: Intelligent advice

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3,585	,453		7,908	,000
	Decision making	,180	,086	,186	2,085	,039

a Dependent Variable: Intelligent advice

Table 4.17: Testing Hypothesis P2

4.4.3 Hypothesis P3

Relation to test: $A4 \rightarrow S5$. As depicted in Table 4.18, Documentation could not significantly account for the variability in agreement to Knowledge Management ($R^2 = .01, R^2_{adj} = .004, F_{(1,120)} = 1.5, p >> .05$). Therefore, this hypothesis is rejected.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,111(a)	,012	,004	,89890

a Predictors: (Constant), Documentation

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1,215	1	1,215	1,504	,222(a)
	Residual	96,964	120	,808		
	Total	98,179	121			

a Predictors: (Constant), Documentation
b Dependent Variable: Knowledge management

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3,842	,385		9,984	,000
	Documentation	,115	,094	,111	1,226	,222

a Dependent Variable: Knowledge management

Table 4.18: Testing Hypothesis P3

4.4.4 Hypothesis P4

Relation to test: $A1 \rightarrow S3$. As depicted in Table 4.19, Communication served as the predictor of Community Building but could not significantly account for the variability in agreement to Community Building ($R^2 = .014, R^2_{adj} = .006, F_{(1,121)} = 1.8, p >> .05$). Therefore, this hypothesis is rejected.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,120(a)	,014	,006	,74352

a Predictors: (Constant), Communication

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	,979	1	,979	1,771	,186(a)
	Residual	66,891	121	,553		
	Total	67,870	122			

a Predictors: (Constant), Communication
b Dependent Variable: Community building

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	3,874	,499		7,758	,000
	Communication	,142	,107	,120	1,331	,186

a Dependent Variable: Community building

Table 4.19: Testing Hypothesis P4

4.4.5 Hypothesis P5

Relation to test: $A3 \rightarrow S2$. Quality Assessment accounted for a significant quantity of the variability in agreement to Search Efficiency ($R^2 = .12, R^2_{adj} = .11, F_{(1,121)} = 16.2, p = .000$). As depicted in Table 4.20, Quality Assessment explained 34% of the variance in Search Efficiency (standardized $\beta = .34, t = 4.0, p = .000$). Therefore, this hypothesis is accepted.

Apparently, retrieving architectural knowledge is important for architects, especially when they are conducting reviews, audits or other quality evaluations. To decide about the quality of an architecture it is important to quickly retrieve the standards, rules and reference architectures that need to be adhered to.

4.4.6 Hypothesis P6

Relation to test: $A3 \wedge A5 \rightarrow S2$. In a multiple linear regression, Quality Assessment and Knowledge Acquisition together accounted for a significant quantity of the variability in agreement to Search Efficiency ($R^2 = .13, R^2_{adj} = .12, F_{(2,120)} = 9.0, p = .000$). We also assessed the relative importance of Quality Assessment and Knowledge Acquisition in predicting the need for Search Efficiency. As depicted in Table 4.21, Quality Assessment was most strongly related to Search Efficiency (30%) (standardized $\beta = .30, t = 3.2, p = .002$). Knowledge Acquisition did not significantly add to this effect (standardized $\beta = .12, t = 1.3, p >> .05$). Supporting this conclusion is the height of the standardized Beta coefficient of

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,343(a)	,118	,110	,57440

a Predictors: (Constant), Quality assessment

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	5,328	1	5,328	16,150	,000(a)
	Residual	39,923	121	,330		
	Total	45,251	122			

a Predictors: (Constant), Quality assessment
b Dependent Variable: Search efficiency

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3,780	,282		13,401	,000
	Quality assessment	,250	,062	,343	4,019	,000

a Dependent Variable: Search efficiency

Table 4.20: Testing Hypothesis P5

Quality Assessment and the strength of the correlation between Quality Assessment and Search Efficiency, partialling out the effects of the other predictor ($r_{partial} = .28, r_{part} = .27$). Knowledge Acquisition offered little or no additional predictive power beyond that contributed by the Quality Assessment measure.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,360(a)	,130	,115	,57282

a Predictors: (Constant), Knowledge acquisition, Quality assessment

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	5,876	2	2,938	8,953	,000(a)
	Residual	39,375	120	,328		
	Total	45,251	122			

a Predictors: (Constant), Knowledge acquisition, Quality assessment
b Dependent Variable: Search efficiency

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients			Correlations		
		B	Std. Error	Beta	t	Sig.	Part	B	Std. Error
1	(Constant)	3,438	,386		8,900	,000			
	Quality assessment	,216	,067	,297	3,213	,002	,343	,281	,274
	Knowledge acquisition	,105	,082	,119	1,291	,199	,234	,117	,110

a Dependent Variable: Search efficiency

Table 4.21: Testing Hypothesis P6

4.5 Prioritization of Support Methods

As a first step in anwsering research question RQ4 we looked at the data of Part 3 of the survey, where the respondents prioritized the five support methods. We used the rank order numbers to calculate an average rank per support option (cf. Table 4.22). We then verified the effects of Block Order (2) on the mean rank order numbers by running a GLM-Repeated Measures for the 5-leveled Ranked Support factor (Ranked Decision Management, Ranked Search Efficiency, Ranked Community Building, Ranked Intelligent Advice, and Ranked Knowledge Management) on the mean ranking. The main effect of Block Order was not significant ($F < 1$) and no significant interaction occurred between Ranked Support and Block Order (Pillai's Trace $= .04, F_{(4,115)} = 1.2, p >> .05$).

Tests of Between-Subjects Effects

Measure: MEASURE_1
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	5398,500	1	5398,500	12453159 17710877 0000,000	,000	1,000
Blockorder	,000	1	,000	,000	1,000	,000
Error	5,12E-014	118	4,34E-016			

Multivariate Tests(b)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Ranked_Support * Blockorder	Pillai's Trace	,041	1,241(a)	4,000	115,000	,298	,041
	Wilks' Lambda	,959	1,241(a)	4,000	115,000	,298	,041
	Hotelling's Trace	,043	1,241(a)	4,000	115,000	,298	,041
	Roy's Largest Root	,043	1,241(a)	4,000	115,000	,298	,041

a Exact statistic
b Design: Intercept+Blockorder
Within Subjects Design: Ranked_Support

Table 4.22: RQ4: Effects of Block order on Ranking

Using the same set up, we also calculated the effects of Organization (4) on the ranking of support options (cf. Table 4.23) but again, the main effect of Organization was insignificant ($F < 1$) and so was the interaction of Organization by Ranked Support (Pillai's Trace $= .13, F_{(12,345)} = 1.3, p >> .05$).

Tests of Between-Subjects Effects

Measure: MEASURE_1
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	2045,704	1	2045,704	.	.
Organization	,000	3	,000	.	.
Error	,000	116	,000		

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.
Ranked_Support * Organization	Pillai's Trace	,129	1,289	12,000	345,000	,223
	Wilks' Lambda	,875	1,291	12,000	299,261	,223
	Hotelling's Trace	,139	1,289	12,000	335,000	,223
	Roy's Largest Root	,099	2,846(b)	4,000	115,000	,027

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+Organization
Within Subjects Design: Ranked_Support

Table 4.23: RQ4: Effects of Organization on Ranking

We did the same for Function (10) but again, the main effect of Function remained insignificant ($F < 1$) and the interaction of Function by Ranked Support as well (Pillai's Trace $= .36, F_{(36,440)} = 1.2, p >> .05$) (cf. Table 4.24). Therefore, in further analyses, data will be pooled over Block Order, Organization, and Function.

Tests of Between-Subjects Effects

Measure: MEASURE_1
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	3555,451	1	3555,451	.	.	1,000
Function	5,97E-016	9	6,63E-017	.	.	1,000
Error	,000	110	,000			

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Ranked_Support * Function	Pillai's Trace	,364	1,225	36,000	440,000	,179	,091
	Wilks' Lambda	,676	1,232	36,000	402,716	,173	,093
	Hotelling's Trace	,422	1,237	36,000	422,000	,169	,095
	Roy's Largest Root	,216	2,636(b)	9,000	110,000	,008	,177

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+Function
Within Subjects Design: Ranked_Support

Table 4.24: RQ4: Effects of Function on Ranking

4.6 Effects of Architecture Activities on the Priority of Support Options

We were now able to study the effects of architecture activities on the priority of support options, in order to answer research question RQ4, which is stated as follows.

- *RQ4: What do architects consider the best tradeoffs between conflicting methods for architectural knowledge sharing support?*

Respondents ranked the support options for one activity alone, so that a 5-leveled between-subjects factor of Activity Ranking Condition could be created (Communication Condition, Decision Making Condition, Quality Assessment Condition, Documentation Condition, Knowledge Acquisition Condition). We ran a GLM-Repeated Measures for the between-subjects factor Activities Ranking Condition and the 5-leveled within-subjects factor of Ranked Support (Ranked Decision Management, Ranked Search Efficiency, Ranked Community Building, Ranked Intelligent Advice, and Ranked Knowledge Management) on the mean rank numbers with Work Years as covariate (cf. Table 4.25).

However, the main effect of Work Years was not significant ($F < 1$) and the interaction between Ranked Support and Work Years also was not significant ($F < 1$). Therefore, we reran the analysis but this time excluding Work Years as covariate. (cf. Table 4.26).

Tests of Between-Subjects Effects

Measure: MEASURE_1
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	1603,607	1	1603,607	38405065 91199205 000,000	,000	1,000
WorkYears	,000	1	,000	,000	1,000	,000
Error	4,76E-014	114	4,18E-016			

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Ranked_Support * WorkYears	Pillai's Trace	,025	,713(a)	4,000	111,000	,585	,025
	Wilks' Lambda	,975	,713(a)	4,000	111,000	,585	,025
	Hotelling's Trace	,026	,713(a)	4,000	111,000	,585	,025
	Roy's Largest Root	,026	,713(a)	4,000	111,000	,585	,025

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+WorkYears+ActivityCondition
Within Subjects Design: Ranked_Support

Table 4.25: Work Years as covariate of 5-leveled within-subjects factor of Ranked Support

The multivariate tests showed that the interaction of Ranked Support by Activity Ranking Condition was not significant ($F < 1$) and that the main effect of Activity Ranking Condition on the mean rank numbers also was not significant ($F < 1$). However, the main effect of Ranked Support did reach significance with a small effect size (Pillai's Trace $= .12, F_{(4,112)} = 3.8, p = .006, \eta_p^2 = .12$). Keeping in mind that the lower rank order number indicates the higher priority, Table 4.27 shows that Ranked Decision Management ($M = 2.63, SD = 1.37$) had higher priority than Ranked Intelligent Advice, which had lowest priority ($M = 3.33, SD = 1.42$).

Multivariate Tests(c)

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Ranked_Support	Pillai's Trace	,120	3,817(a)	4,000	112,000	,006	,120
	Wilks' Lambda	,880	3,817(a)	4,000	112,000	,006	,120
	Hotelling's Trace	,136	3,817(a)	4,000	112,000	,006	,120
	Roy's Largest Root	,136	3,817(a)	4,000	112,000	,006	,120
Ranked_Support * ActivityCondition	Pillai's Trace	,112	,829	16,000	460,000	,653	,028
	Wilks' Lambda	,891	,825	16,000	342,803	,657	,028
	Hotelling's Trace	,119	,821	16,000	442,000	,661	,029
	Roy's Largest Root	,079	2,284(b)	4,000	115,000	,065	,074

a Exact statistic
b The statistic is an upper bound on F that yields a lower bound on the significance level.
c Design: Intercept+ActivityCondition
Within Subjects Design: Ranked_Support

Table 4.26: Multivariate tests (without workyears as covariate)

	Activity for which support options were ranked	Mean	Std. Deviation	N
Ranked Decision Management	Communication	2,78	1,166	23
	Decision Making	2,54	1,474	24
	Quality Assessment	2,75	1,351	28
	Documentation	2,76	1,338	21
	Knowledge Acquisition	2,29	1,517	24
	Total	2,63	1,366	120
Ranked Search Efficiency	Communication	2,91	1,379	23
	Decision Making	2,88	1,329	24
	Quality Assessment	2,96	1,232	28
	Documentation	2,57	1,287	21
	Knowledge Acquisition	2,92	1,248	24
	Total	2,86	1,279	120
Ranked Community Building	Communication	3,39	1,559	23
	Decision Making	2,88	1,541	24
	Quality Assessment	3,04	1,347	28
	Documentation	3,90	1,480	21
	Knowledge Acquisition	3,33	1,659	24
	Total	3,28	1,529	120
Ranked Intelligent Advice	Communication	3,30	1,579	23
	Decision Making	3,38	1,173	24
	Quality Assessment	3,50	1,622	28
	Documentation	3,05	1,359	21
	Knowledge Acquisition	3,33	1,373	24
	Total	3,33	1,421	120
Ranked Knowledge Management	Communication	2,61	1,340	23
	Decision Making	3,33	1,494	24
	Quality Assessment	2,75	1,481	28
	Documentation	2,71	1,347	21
	Knowledge Acquisition	3,13	1,076	24
	Total	2,91	1,366	120

Table 4.27: Descriptive Statistics for the Effects of Activities on Ranked Support

Paired-samples t-tests showed that five out of 10 comparisons among the Ranked Support options were significant, which means that the need for support is a sliding scale of priorities (cf. Table 4.28). Table 4.30 depicts the paired-samples t-tests for comparisons between priorities of support options (df = 119).

In combining Table 4.27 and Table 4.28, we get to a hierarchy of support options that are more or less needed for architecting activities. The lower rank numbers indicate the higher priority and the sign means that there is no significant difference between the mean rank numbers of two support options. According to these architects, Decision Management has highest priority and Intelligent Advice lowest, irrespective of the activities they are involved in.

Priority of need for support throughout all architecture activities:

- **Most needed:** Decision Management (M = 2.63) ≅ Search Efficiency (M = 2.86)
- **Needed:** ≅ Knowledge Management (M = 2.91) ≅ Community Building (M = 3.28)
- **Least needed:** ≅ Intelligent Advice (M = 3.33)

	t	*p*
Ranked Decision Management - Ranked Search Efficiency	-1.2	.221
Ranked Decision Management - Ranked Community Building	-3.0	.003**
Ranked Decision Management - Ranked Intelligent Advice	-3.6	.000**
Ranked Decision Management - Ranked Knowledge Management	-1.5	.146
Ranked Search Efficiency - Ranked Community Building	-2.1	.038*
Ranked Search Efficiency - Ranked Intelligent Advice	-2.3	.023*
Ranked Search Efficiency - Ranked Knowledge Management	-.28	.777
Ranked Community Building - Ranked Intelligent Advice	-.20	.841
Ranked Community Building - Ranked Knowledge Management	1.7	.085
Ranked Intelligent Advice - Ranked Knowledge Management	2.0	.048*

Table 4.28: Ranking of Support Methods

Tests of Between-Subjects Effects

Measure: MEASURE_1
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	5353,448	1	5353,448	10311950091815000000,000	,000	1,000
ActivityCondition	,000	4	,000	,000	1,000	,000
Error	5,97E-014	115	5,19E-016			

Table 4.29: Tests of Between-Subjects Effects

Paired Samples Test

		Paired Differences						df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Upper	Lower	t	Std. Deviatio n	St d. Er ror M ea n
Pair 1	Ranked Decision Management - Ranked Search Efficiency	-,233	2,077	,190	-,609	,142	-1,231	119	,221
Pair 2	Ranked Decision Management - Ranked Community Building	-,658	2,399	,219	-1,092	-,225	-3,006	119	,003
Pair 3	Ranked Decision Management - Ranked Intelligent Advice	-,700	2,117	,193	-1,083	-,317	-3,623	119	,000
Pair 4	Ranked Decision Management - Ranked Knowledge Management	-,283	2,123	,194	-,667	,100	-1,462	119	,146
Pair 5	Ranked Search Efficiency - Ranked Community Building	-,425	2,218	,202	-,826	-,024	-2,099	119	,038
Pair 6	Ranked Search Efficiency - Ranked Intelligent Advice	-,467	2,223	,203	-,868	-,065	-2,300	119	,023
Pair 7	Ranked Search Efficiency - Ranked Knowledge Management	-,050	1,931	,176	-,399	,299	-,284	119	,777
Pair 8	Ranked Community Building - Ranked Intelligent Advice	-,042	2,273	,207	-,453	,369	-,201	119	,841
Pair 9	Ranked Community Building - Ranked Knowledge Management	,375	2,362	,216	-,052	,802	1,739	119	,085
Pair 10	Ranked Intelligent Advice - Ranked Knowledge Management	,417	2,288	,209	,003	,830	1,995	119	,048

Table 4.30: Paired Samples Test

Chapter 5

Discussion

The survey results offer good food for thoughts. Although we have found some plausible explanations for our findings, we were quite interested in the opinion of practicing architects. Therefore, we asked the respondents of our survey to provide feedback on the results. We sent them a bullet-wise highlight report by email that summarized the main results that were presented in the previous section. This email triggered several responses within days, from which we could distill a number of interesting quotes. Some of these quotes are included in the discussion below.

We were not really surprised to see that Section 4.1.1 showed that block order was insignificant for Part 1 and Part 2 of our survey. This allowed us to include all respondents (who were spread over 10 versions of the survey) in our further analyses. We were, also not really surprised to see that the type of Function and Organization also did not lead to significantly different results. Apparently the participating organizations were homogeneous and the various architects – including IT, solution, software, and enterprise architects – all work more or less on the same activities and also have similar preferences for support methods for sharing architectural knowledge.

With respect to what architects do, the most striking result is that architects seem to make lots of architectural decisions, but – especially the less experienced ones – neglect documenting those decisions and their rationale. Most obvious reasons for this is a lack of time or interest. Often the benefits of documenting design rationale are only visible in the longer term; in the short term the most important thing is to meet the deadline, and move on to the next project. The lack of interest in long-term knowledge reuse is in line with observations from Harrison et al., who found that architects lack a real motivation to document and maintain architectural knowledge [19].

The lack of a defined, visible process for architecting activities could also partly explain why architects omit to document much. If the role, responsibilities, and extent of the authority of architects would be better defined, these architects can be made more accountable by the organization for their actions. Experience in several software projects shows that a well-defined, visible process (e.g, by using a charter) stimulates architects to document more knowledge [28].

Another reason for not documenting architectural knowledge is that it is already known to the architect, who possesses it as tacit knowledge in his mind, so for this architect no direct need of making this knowledge explicit exists. This phenomenon is acknowledged by Kruchten et al., who also mention that as a consequence, architects cannot revisit or communicate these decisions. [30]. Bosch also acknowledges this problem, which he called knowledge vaporization [3]. This process is to be prevented at all costs, because if the reasoning behind an architectural solution is lost or 'forgotten', violations to the architecture remain undetected, causing design erosion.

Two other reasons for not documenting design decisions were given by architects who responded to our feedback request:

> *"Whether or not to document is a tough decision. Doing so could make yourself redundant in case questions are asked later on, and that is what worries architects most."*

> *"Many architects have a technical background and were programmers, designers or system maintainers before they became an architect. For these people, documenting has always been a pain, and this is a mindset they still have."*

Our results are in line with a study on the architect's mindset by Clerc et al. [7]. They conducted a survey among architects in the Netherlands and found that the prevalent mindset of architects reveals an approach focused on 'to create and communicate' rather than 'to review and maintain' an architecture. In our study decision making and communication scored highest and quality assessment and documentation scored lowest, which further underlines the emphasis put on creating short-term solutions, rather than maintaining a body of knowledge that is useful on the longer term as well.

When looking at the prioritization of support methods we were not very surprised to see that managing architectural design decisions was found most important. This is in line with an increasing focus on design decisions in the software architecture research community, as reflected in the formulation of a decision view on software architecture practice [30] and the adoption of central concepts as 'decision' and 'rationale' to the upcoming ISO/IEC 42010 standard for software architecture description [23].

The interesting thing, however, is that our results show that experienced architects differ slightly in their mindset; experienced architects do more on quality assessments and documentation. Perhaps they value these activities higher because they have encountered more situations in which sufficient documentation proved to be useful. Another explanation could be that writing architectural documentation is easier if you are more experienced (e.g., formulating rationale for decisions, or drawing effective architectural views). Likewise, conducting architectural audits is something that is only possible if you have sufficient experience in the field yourself. This makes it easier to spot inconsistencies, conflicts, or missing qualities of the system.

As for the other support methods, we were surprised that 'knowledge management' support received the lowest scores (cf. Section 4.3) and that 'intelligent advice' was ranked lowest by the architects (cf. Section 4.4). These two categories both alleviate architects in their routine, knowledge processing tasks by automatically retrieving, and manipulating information, offering suggestions, etc. Especially since architects appear to have little time or interest in documenting architectural knowledge, we had expected that all pro-active support to help them with this would be more appreciated, and make certain basic tasks less cumbersome.

We suspect architects are a bit frightened by the idea that smart automated tooling would take over their role, although they acknowledge the positive effects (pattern P2 was significant). Architects want to sit in the driver's seat and have a tight control over all processes. That also explains why support methods that do not endanger this role (e.g., search efficiency and decision management) are clearly ranked higher. Another reason for the relative low rank of intelligent support could be the lack of effective implementations of such support. We heard from several architects that most intelligent tools and methods are still rather immature and add little to their daily work, but that more versatile and mature implementations would be welcomed:

> *I find most specialized tools to have too little functionality that supports my work as a software architect. As a result, I often resort to more generic tools such as MS Visio, spreadsheets, etc."*

The fact that architects want to have control over their actions is further underlined by the fact that in the prioritization of support methods, community building is ranked fourth (out of five). We had expected the architects to be a bit more 'team-players', or at least more in favor of creating a 'community of architects' in which experts effectively exchange knowledge, discuss experiences, and build up a collective memory. Our study gives us the impression that practicing architects are rather 'lonesome'. They are not the community builders we had expected them to be. Instead they act rather in splendid isolation. Several architects confirmed this, two of whom explained it as follows:

> *"I recognize myself in the 'lonesome architect' characterization and find it actually quite logical. An architect is often busy working in the early stages of the software life cycle and of projects. Colleagues are busy helping to solve problems of other customers and thus not around. Furthermore, there is often no guidance for the work you do as an architect because the systems you help design do not exist yet."*

> *"An architect usually considers his own judgment as most precise and valuable. Referring to others (humans or tools) is often only a strategic move."*

Coming back to our main research question, our study provides valuable insight into what architects do and what support they need for sharing architectural knowledge. Architects are individual experts who consume substantial amounts of architectural knowledge but care less for actively sharing such knowledge. We conjecture that the best way to motivate these lonesome architects in sharing architectural knowledge involves non-intrusive support that offers various mechanisms to easily store, communicate or manipulate knowledge in the architecting process. Based on the results of our survey, we postulate that such support should have two fundamental characteristics:

1. **It respects the architect's autonomy.** We learned that architects like to control the processes they are involved in. Methods that support

architectural knowledge sharing should respect this. Automated support that 'thinks' for the architect, or tools that prescribe architects what to do are not going to work. What does work, however, are more person-centric descriptive tools that assist knowledge workers 'on the fly' during daily routines [8]. Although in this survey architects ranked intelligent support as least needed, feedback obtained suggests that such support could still be valuable if implementations would be sufficiently versatile and mature.

2. **It stimulates both production and consumption of architectural knowledge.** We saw that support for decision management and search efficiency were most needed according to practicing architects. Search efficiency support assists the autonomous, or lonesome, architect in retrieving relevant architectural knowledge when needed, and is thus oriented towards consumption of architectural knowledge. Decision management helps architects to manage their thought processes when taking (new) design decisions, solving conflicts, or codifying design rationale. This kind of support thus focuses mostly on the creation – or production – of architectural knowledge. Based on our survey results, we argue that mature support for sharing architectural knowledge should assist architects in both producing and consuming such knowledge.

   A promising development in software architecture is the emergence of integrated platforms for sharing architectural knowledge that implement a range of 'use cases' related to production and consumption of knowledge. A state-of-the-art review of such platforms is provided by Liang and Avgeriou [34]. A main characteristic of some of these platforms is that they follow a hybrid strategy for architectural knowledge sharing [14]. As Lago et al. argue, such a strategy "support[s] the knowledge producers in efficiently documenting the knowledge and the consumers in using it" [33], which would resolve the imbalance of producing and consuming knowledge we observed during our study.

As a final note, we stress that the identification of architectural knowledge sharing patterns and subsequent development of appropriate methods and tools does not automatically lead to increased knowledge sharing behavior among architects. Lots of other influencing or motivational factors exist that induce architects to share knowledge [18], including personal experience, skills and background, but also organizational context, culture and beliefs. This is acknowledged by Schneider, who argues that "It is a major misunderstanding to expect altruistic behavior. An expert or user needs reasons to provide input". [38]. Apart from offering effective support methods, organizations can do a lot to stimulate knowledge sharing. Some obvious ones include setting up organizational activities (working in small competence teams, helping each other with producing deliverables) and communication activities (organize workshops and brainstorm sessions) in order to motivate architects to share knowledge with peers. Our study shows that real intrinsic motivation, however, probably involves all-round support that allows architects to stay in control of all his architectural knowledge assets.

5.1 Threats to Validity

We list possible limitations to our study by discussing the internal validity, construct validity and external validity following [27] and [37]. Our questionnaire design and pilot study ensured that our questions were unambiguous, to-the-point and our hypotheses well-focused on the research question we wanted to answer. This made analysis of the survey data relatively easy, and increased the chance for meaningful and significant results. With respect to *internal validity*, our questionnaire design leaves little room for selection bias or confounding variables.

To conform to *construct validity* we constructed our theory (cf. Figure 2.1) as one of the first steps of our survey design, based on which the items of the questionnaire were phrased. During both the pilot study and main study we conducted scale analysis and factor analysis to assess whether our theorized concepts could be unambiguously retrieved in the data. In the pilot study this helped us to refactor our original framework (as presented in [15]) into the framework presented in this paper. Scale and factor analysis during the main study indicated that indeed five factors of architecting activities and support methods could be identified, which indicates good construct validity.

With respect to *external validity* we deem our results fairly generalizable. Our sample was relatively large, and the fact that four different organizations participated decreased the chance for organizational bias significantly. To obtain a fair reflection of a typical architecting process, many types of architects were included in our study, including software, IT, solution, enterprise and infrastructure architects. However, even though the participating companies were international, our study only focused on practicing architects working in the Netherlands. Our study thus does not deal with possible cultural or educational factors in which Dutch architects differ from their colleagues.

Chapter 6

Conclusions

Architects are complicated individuals. They make lots of architectural decisions, but neglect to document them. Producing and subsequently sharing of architectural knowledge is clearly not one of their most popular activities. For this reason, one would expect that supporting architects in this latter task is something they appreciate, but the opposite is true. Architects rather stay in control themselves and still need to be convinced of the value of current automated or intelligent support. When it comes to consumption of architectural knowledge, however, architects indicate that support for effective retrieval of (stored) architectural knowledge is on the top of their wish list. This imbalance in production and consumption shows that with respect to sharing architectural knowledge architects are not the 'community builders' many expect them to be. Instead, they are rather lonesome decision makers who act in splendid isolation.

Somehow the results of our study remind us of a well-known song:.

> *['Lonesome Cowboy', by Elvis Presley]*
>
> *I am just a lonesome cowboy*
> *And Im travelling all alone*
> *.....*

Our survey research helped unraveling what architects really do and what they need with respect to sharing architectural knowledge. The results act as call for awareness to both researchers and practitioners. We found that effective support for sharing architectural knowledge should acknowledge 'the nature of the beast'. We therefore plead for descriptive, non-intrusive support that respects the architect's autonomy. Moreover, to be effective, this support should stimulate architects in both producing and consuming architectural knowledge. Integrated, 'all-round', tools environments that implement a variety of architectural knowledge sharing use cases, seem promising because they could provide a solid balance between production and consumption of knowledge in the architecting process.

As future work we plan to look more closely at preferences for support methods with more emphasis on tradeoffs. This allows us to gain a better understanding of what architects prefer if they have to explicitly choose between support methods (RQ4). The combination of ties and significant differences between support options (cf. Section 4.6) makes it worthwhile to perform a conjoint analysis based on the rank order data [36].

Bibliography

[1] Muhammad Ali Babar and Ian Gorton. A Tool for Managing Software Architecture Knowledge. In *2nd Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI)*, Minneapolis, USA, 2007.

[2] P. Avgeriou, P. Kruchten, P. Lago, Paul Grisham, and Dewayne Perry. Architectural Knowledge and Rationale - Issues, Trends, Challenges. *ACM SIGSOFT Software Engineering Notes*, 32(4):41–46, 2007.

[3] Jan Bosch. Software Architecture: The Next Step. In *1st European Workshop on Software Architectures (EWSA)*, pages 194–199, St. Andrews, UK, 2004.

[4] Rafael Capilla, Francisco Nava, Sandra Pérez, and Juan C. Dueñnas. A Web-based Tool for Managing Architectural Design Decisions. In *1st ACM Workshop on SHaring ARchitectural Knowledge (SHARK)*, Torino, Italy, 2006.

[5] Rafael Capilla, Francisco Nava1, and Juan C. Dueñas. Modeling and Documenting the Evolution of Architectural Design Decisions. In *2nd Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI)*, Minneapolis, USA, 2007.

[6] Paul Clements, Rick Kazman, Mark Klein, Divya Devesh, Shivani Reddy, and Prageti Verma. The Duties, Skills, and Knowledge of Software Architects. In *6th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 2007.

[7] Viktor Clerc, Patricia Lago, and Hans van Vliet. The Architectś Mindset. In *3rd International Conference on the Quality of Software-Architectures (QoSA)*, pages 231–249, Boston, USA, 2007.

[8] Kathryn Cormican and Lawrence Dooley. Knowledge Sharing in a Collaborative Networked Environment. *Journal of Information & Knowledge Management*, 6(2):105–114, 2007.

[9] Bredemeyer D. and Malan R. The Role of the Software Architect. Technical Report White paper 12/10/04, Bredemeyer Consulting, 2002.

[10] Peter Eeles. Characteristics of a Software Architect. Technical Report Available online, 2006.

[11] Weiguo Fan, Linda Wallace, Stephanie Rich, and Zhongju Zhang. Tapping the Power of Text Mining. *Communications of the ACM*, 49(9):77–82, 2006.

[12] Rik Farenhorst and Remco C. de Boer. Knowledge Management in Software Architecture: State of the Art. In Muhammad Ali Babar, Torgeir Dingsøyr, Patricia Lago, and Hans van Vliet, editors, *Software Architecture Knowledge Management: Theory and Practice*. Springer, 2009.

[13] Rik Farenhorst, Patricia Lago, and Hans van Vliet. EAGLE: Effective Tool Support for Sharing Architectural Knowledge. *International Journal of Cooperative Information Systems (IJCIS)*, 16(3/4):413–437, 2007.

[14] Rik Farenhorst, Patricia Lago, and Hans van Vliet. A Just-In-Time Architectural Knowledge Sharing Portal. In *7th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 2008.

[15] Rik Farenhorst and Hans van Vliet. Understanding How to Support Architects in Sharing Knowledge. In *4th Workshop on SHAring and Reusing architectural Knowledge (SHARK)*, Vancouver, Canada, 2009.

[16] Rik Farenhorst and Hans van Vliet. Experiences with a Wiki to Support Architectural Knowledge Sharing. In *3rd Workshop on Wikis for Software Engineering (Wikis4SE)*, Porto, Portugal, —2008—.

[17] David Garlan and Bradley Schmerl. The RADAR Architecture for Personal Cognitive Assistance. *International Journal of Software Engineering and Knowledge Engineering*, 17(2), 2007.

[18] Tanu Ghosh. Creating Incentives for Knowledge Sharing. Technical report, MIT Open Courseware, Sloan school of management, Cambridge, Massachusetts, USA, 2004.

[19] N.B. Harrison, Paris Avgeriou, and U. Zdun. Using Patterns to Capture Architectural Decisions. *IEEE Software*, 24(4):38–45, 2007.

[20] C. Hofmeister, P. Kruchten, Robert Nord, Henk Obbink, Alexander Ran, and Pierre America. A General Model of Software Architecture Design Derived from Five Industrial Approaches. *The Journal of Systems and Software*, 80(1):106–126, 2007.

[21] Johan F. Hoorn, Mark E. Breuker, and Evelien Kok. Shifts in Foci and Priorities. Different Relevance of Requirements to Changing Goals Yields Conflicting Prioritizations and is Viewpoint-dependent. *Software Process Improvement and Practice*, 11:465–485, 2006.

[22] Marleen Huysman and Dirk de Wit. Practices of Managing Knowledge Sharing: Towards a Second Wave of Knowledge Management. *Knowledge and Process Management*, 11(2):81–92, 2004.

[23] ISO/IEC. Systems and Software Engineering – Architecture Description. Standard ISO/IEC 42010, 2009.

[24] A. Jansen, Jan S. van der Ven, P. Avgeriou, and Dieter K. Hammer. Tool Support for Architectural Decisions. In *6th Working IEEE/IFIP Conference on Software Architecture*, Mumbai, India, 2007.

[25] Anton Jansen and Jan Bosch. Software Architecture as a Set of Architectural Design Decisions. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 109–120, Pittsburgh, USA, 2005.

[26] B. Kitchenham and S.L. Pfleeger. Principles of Survey Research, Parts 1 to 6. *Software Engineering Notes*, 2001-2002.

[27] Barbara A. Kitchenham, Shari Lawrence Pfleeger, David C. Hoaglin, and Jarrett Rosenberg. Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE transactions on Software Engineering*, 28(8), 2002.

[28] Philippe Kruchten. The Architects - The Software Architecture Team. In *1st Working IFIP Conference on Software Architecture (WICSA)*, San Antonio, USA, 1999.

[29] Philippe Kruchten. What do software architects really do? *The Journal of Systems and Software*, 81:2413–2416, 2008.

[30] Philippe Kruchten, Rafael Capilla, and Juan Carlos Dueñnas. The Decision View's Role in Software Architecture Practice. *IEEE Software*, 26(2):36–42, 2009.

[31] Philippe Kruchten, Patricia Lago, and Hans van Vliet. Building up and Reasoning about Architectural Knowledge. In *2nd International Conference on the Quality of Software Architectures (QoSA)*, pages 39–47, Stockholm, Sweden, 2006.

[32] Patricia Lago and Paris Avgeriou. 1st Workshop on SHaring and Reusing ARchitectural Knowledge, Final Workshop Report. *ACM SIGSOFT Software Engineering Notes*, 31(5):32–36, 2006.

[33] Patricia Lago, Paris Avgeriou, Rafael Capilla, and Philippe Kruchten. Wishes and Boundaries for a Software Architecture Knowledge Community. In *7th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, page 4, Vancouver, Canada, 2008.

[34] Peng Liang and Paris Avgeriou. Tools and Technologies for Architectural Knowledge Management. In Muhammad Ali Babar, Torgeir Dingsøyr, Patricia Lago, and Hans van Vliet, editors, *Software Architecture Knowledge Management: Theory and Practice*. Springer, 2009.

[35] Peng Liang, Anton Jansen, and Paris Avgeriou. Collaborative Software Architecting through Knowledge Sharing. In I. Mistrik, editor, *Collaborative Software Engineering*. Springer, 2009.

[36] Bryan K. Orme. *Getting Started with Conjoint Analysis: Strategies for Product Design and Pricing Research*. Research Publishers LLC, 2006.

[37] Dewayne Perry, Adam A. Porter, and Lawrence Votta G. Empirical studies of software engineering: a roadmap. In *Conference on The Future of Software Engineering*, pages 345–355, Limerick, Ireland, 2000.

[38] Kurt Schneider. Modeling and Improving Information Flows in the Development of Large Business Applications. In Muhammad Ali Babar, Torgeir Dingsøyr, Patricia Lago, and Hans van Vliet, editors, *Software Architecture Knowledge Management: Theory and Practice*. Springer, 2009.

[39] Jeff Tyree and Art Akerman. Architecture Decisions: Demystifying Architecture. *IEEE Software*, 22(2):19–27, 2005.

[40] Bas van der Raadt, Sander Schouten, and Hans van Vliet. Stakeholder Perception of Enterprise Architecture. In *2nd European Conference on Software Architecture (ECSA)*, volume LNCS 5292, pages 19–34, Paphos, Cyprus, 2008. Springer.