The Lonesome Architect

Rik Farenhorst[1], Johan F. Hoorn[2], Patricia Lago[1], Hans van Vliet[1]
[1]Department of Computer Science, VU University Amsterdam, the Netherlands
[2]Center for Advanced Media Research Amsterdam, VU University Amsterdam, the Netherlands
[1]{rik,patricia,hans}@cs.vu.nl, [2]jf.hoorn@camera.vu.nl

Abstract

Although the benefits are well-known and undisputed, sharing architectural knowledge is not something architects automatically do. In an attempt to better understand what architects really do and what kind of support they need for sharing knowledge, we have conducted large-scale survey research. The results of our study indicate that architects can be characterized as rather lonesome decision makers who mainly consume, but neglect documenting and actively sharing architectural knowledge. Acknowledging this nature of architects suggests ways to develop more effective support for architectural knowledge sharing.

1 Introduction

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, 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 [19, 26, 32]. 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 [1,4,13,20], 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].

In this paper we assess to what extent the above claims hold, by unraveling what architects do and what kind of support they need. Consequently, our main research question is: *What do architects do and what kind of support do they need for sharing architectural knowledge?*

We have constructed a theory on architecting, based on a review of state-of-the-art literature 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 the existence of a number of 'patterns': certain methods for sharing architectural knowledge are more supportive than others to architects involved in a specific activity. The identification of such patterns helps in the development and maturation of methods and tools for sharing architectural knowledge.

To validate our theory we have conducted 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 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.

978-1-4244-4985-9/09/ 25.00

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 [28]. 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 [10]. 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, combined with established literature on software architecture and knowledge management, we developed a theory to characterize what architects do and what they need for sharing architectural knowledge. In the next three subsections 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).

2.1 Architecting Activities

Our theory of architecting activities is depicted in Table 1 and consists of five categories: communication, decision making, quality assessment, documentation, and knowledge acquisition. For each activity we defined several subactivities that are most indicative to this activity, which was tested by our online survey. The remainder of this section highlights some of the related work our theory was built on.

According to Philippe Kruchten, software architects should "make design choices, validate them, and capture them in various architecture related artifacts" [24]. Doing all these things involves a lot of consensus *decision making* in which architects balance between quality attributes, stakeholder concerns, and requirements, and in which they try to apply architectural styles and patterns.

A study by Clements et al. on the duty, knowledge, and skills of architects confirms the above view of the primary tasks of architects [6]. They 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 optimal solutions [13]. As input to this process, architects need a vast body of architectural knowledge. According to Clements et al. this body of knowledge includes basic computer science knowledge, design knowledge 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.

Table 1. Architecting Activities

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.

Another main architecting activity is to *document* architectural knowledge. This includes not only 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 or 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, 27]. 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.

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. Table 2 shows our theory of support methods for sharing architectural knowledge (i.e. decision management, search efficiency, community building, intelligent advice, and knowledge management). The indicative sub-methods listed in this table served as items on our survey.

Table 2. Support Methods

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.

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 managing concepts like design decisions, rationale, and related architectural knowledge [1, 5, 20]. Support for management of design decisions could help architects involved in decision making. While working on the 'backlog' of open issues and challenges they can organize their thought processes and manage the architectural knowledge they produce [16].

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 [27]. 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 [12]. Support for *efficient searching* of architectural knowledge will stimulate reuse (reusable assets are better accessible) and 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 an introduction to how a collaborative architecting process would look like [30]. 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 improves 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 [14].

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, such as the Knowledge Architect tool suite that, among other things, assists architects in writing documentation, and that can perform quantitative architectural analysis [30].

The last type of support we envision is advanced *management of architectural knowledge* that includes refinement, enrichment and smart overviews of such knowledge. Architects could benefit from (semi)-automatic interpretation of stored knowledge to enrich it. Text mining services (e.g., [9]) 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 and 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.

2.3 Architectural Knowledge Sharing Patterns

Table 1 and Table 2 show the factors in our theory on architecting activities and support for architectural knowledge sharing. We hypothesized that specific support might be more 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, as depicted in Figure 1. For each pattern we formulated a hypothesis that was tested with our survey. As a straightforward example, pattern P1 states that decision making (A2) demands decision management support (S1). Pattern P2 claims that architects busy with taking architectural decisions (A2) also want 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. 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.

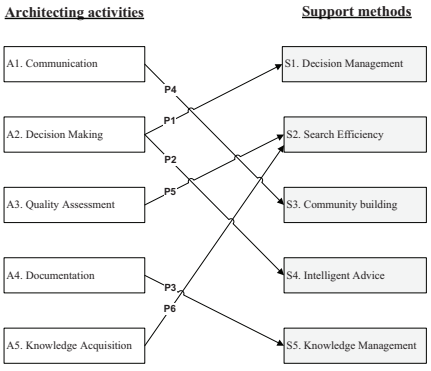

Figure 1. Hypothesized Patterns of Architectural Knowledge Sharing Support

3 Research Methodology

In total 279 Dutch architects from four IT organizations were included in our study. Three organizations were international software development and consultancy firms in the public, finance and industry domains. The fourth one was a central Dutch governmental organization for the governance of ICT architectures in the public domain.

Our research methodology includes all important elements required to properly design, administer and analyze a survey [21]. Due to space limitations in the remainder of this paper we only highlight key aspects of our survey design and analysis. For a complete discussion we refer to the technical report [11].

Our study consisted of a pilot study and a main study, which are further explained below.

3.1 Pilot Study

The theoretical framework presented in Section 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. 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. To give

an example, in our earlier theory we made a distinction between communication within a team (with colleagues) and communication outside the team (with other stakeholders). However, the factor analysis of the pilot questionnaire indicated that this distinction is not made by architects; the items of both constructs all mapped on one factor. Consequently, we merged these two constructs and labeled the new one 'Communication'. This is why our final theory of architecting activities (Table 1) contained five categories and not six. In a similar fashion, we refactored our theory of support methods. Coincidentally, again we had to change the theory by going from six to five constructs. For each of these constructs the five or six most explanatory items were selected, as depicted in Table 2.

3.2 Main Study

In our main study we tested our theoretical framework as presented in Section 2. To properly answer our main research question, which involved unraveling what architects do and what support they need, the questionnaire of our main study consisted of three parts:

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 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).

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. 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'*

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.

We split up the sample in 5 equally large homogeneous groups of architects. For these groups Part 1 and Part 2 of the questionnaire was identical. Part 3, however, only contained *one* prioritization question to prevent bias and fatigue effects [17]. Each respondent thus prioritized the support methods for just one architecting activity. Further details on our questionnaire design are found in [11].

In addition to the questions of the three parts, our (anonymous) questionnaire contained some additional questions including the respondent's job title (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 appreciate (Part 2) and their prioritization of those support methods (Part 3).

To analyze the survey data we used a range of analysis techniques, as discussed in-depth in [11].

4 Survey 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. From this group, 13 were discarded in the analysis because their records were completely empty. 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%.

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 architecture-related working experience was 6 years; 26 architects indicated less than 3 years of architecting experience, 89 architects 5 years or more.

To assess whether our data was internally consistent and the items were really indicative for the constructs of our framework, we conducted scale analysis. Factor analysis indicated that all five constructs of activities (Part 1) and five constructs of support methods (Part 2) could be retrieved in the data. Regular Cronbach's alphas were between .61 and .90, indicating a reasonable to good reliability.

4.1 What Architects Do

To find out what practicing architects really do we studied the data of Part 1 of our questionnaire. As depicted in Table 3, the activity Decision Making received the highest agreement scores ($\mu = 5.21, \sigma = .65$) and Documentation the lowest ($\mu = 3.98, \sigma = .89$). Analysis of variance showed that these results were not significantly different between the four participating organizations, nor did the type of architect impact this outcome. We did, however, find a significant correlation between the amount of work experience architects have and the activities they are involved in. Particularly, agreement to Quality Assessment and Documentation increased significantly with the number of Work Years (see [11]).

Table 3. Descriptive Statistics For Activities and Support Methods

	Mean	Std. Deviation	Correlation with Work Years	N
Work years (experience in architecture projects)	*6.22*	*4.110*		*142*
Architecting activities				
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
Support methods				
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

Our survey results thus indicate that interestingly enough architects spend most of their time on making architectural decisions and least on documenting the results. This runs the risk of leading to substantial architectural knowledge vaporization [3] and substantially decreases the chances for effectively reusing architectural knowledge.

The effect of work experience indicates that more experienced architects seem more often involved in auditing activities, and – maybe related to this – spend more energy on documenting their results. Maybe the advantages of retrieving or reusing codified knowledge has proven itself to these experienced architects over the years.

4.2 What Architects Need

To examine what kind of support methods architects desire, we studied the data of Part 2 of our questionnaire. As depicted in Table 3 Search Efficiency raised the highest agreement scores ($\mu = 4.89, \sigma = .61$) and Knowledge Management the lowest ($\mu = 4.30, \sigma = .90$). Again, analysis of variance showed no significant differences between types of architects or between architects from different organizations. The interaction between Support and Work Years also was not significant. Search Efficiency and Decision Management were equally agreed upon as the kind of support most needed, irrespective of Work Years.

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 that architects really wish to use codified knowledge, but refrain from codifying it in the first place. A proper balance between producing and consuming architectural knowledge, as deemed important in [27], is lacking.

Another surprise related to architects' opinion about the various support methods is 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.

4.3 Preferences in Support Methods

We examined which support methods architects prefer by conducting two separate analyses: a) we tested the patterns hypothesized in Figure 1 by trying to retrieve them in the data of Part 1 and Part 2 of our questionnaire, and b) we examined the ranking data of Part 3.

For our first analysis we formulated six regression models in which the level of agreement to activities should predict the level of agreement to the kind of support needed (see Figure 1): $A1 \rightarrow S3$, $A2 \rightarrow S1$, $A2 \rightarrow S4$, $A3 \rightarrow S2$, $A4 \rightarrow S5$, $A3\&A5 \rightarrow S2$. Linear regression analysis confirmed the hypotheses formulated for patterns P1, P2, and P5. In each case the corresponding activity explained to a considerable extent the variance in agreement to the specific support methods:

P1 Decision Making explained 25% of the variance in Decision Management (P1). 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.

P2 Decision Making explained 19% of the variance in Intelligent Advice (P2). 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 particularly helpful for architects who are busy taking design decisions.

P5 Quality Assessment explained 34% of the variance in Search Efficiency (P5). Retrieving architectural knowledge is important for architects, particularly when they are conducting audits or other quality evaluations. Apparently, to decide about the quality of an architecture it is important to quickly retrieve the standards, rules, etc. that need to be adhered to.

The other three patterns were rejected, because in these cases the activities did not significantly explain the variance

in agreement to the corresponding support methods. Apparently, advanced Knowledge Management support is not particularly important to architects busy with Documentation (P3), Community Building is not particularly interesting to architects active in Communication (P4), and Search Efficiency is not particularly appreciated by architects busy with Knowledge Acquisition (P6).

For our second analysis we looked at the data of Part 3 of the survey, where the respondents ranked the five support methods. Analysis of variance showed no significant differences between the five groups over which the respondents had been divided. Apparently, architects prioritize the five support methods regardless of the activity they are involved in. This means we can aggregate the results of the five groups, as depicted in Table 4 by the 'Total' rows.

Keeping in mind that the lower rank order number indicates the higher priority, Table 4 shows that Decision Management ($\mu = 2.63, \sigma = 1.37$) had higher priority than Intelligent Advice, which had lowest priority ($\mu = 3.33, \sigma = 1.42$). Further data analysis showed that the need for support is a sliding scale. In other words, our survey indicated a hierarchy of support methods that are more *or* less needed for architecting activities. According to the architects, Decision Management had highest priority and Intelligent Advice lowest, irrespective of the activities they were involved in. Interpreting these results allows us to group the support methods as follows:

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

The lower rank numbers indicate the higher priority and the $\cong$ sign means that there is no significant difference between the mean rank numbers of two support methods, according to a t-test.

Table 4. Rankings of Support Methods

	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**

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.

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 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 [23].

Another reason for not documenting architectural knowledge is that architects already possess this knowledge in their minds. There is no direct need for making this tacit knowledge explicit. This phenomenon is acknowledged by Kruchten et al., who also mention that as a consequence, architects cannot revisit or communicate these decisions [25]. 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', it may cause design erosion.

Two more reasons for not documenting were provided 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 is 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 for the longer term as well.

When looking at the prioritization of support methods we were not 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 [25] and the adoption of central concepts as 'decision' and 'rationale' to the upcoming ISO/IEC 42010 standard for software architecture description [18].

As for the other support methods, we were surprised that 'knowledge management' support received the lowest scores (cf. Table 3) and that 'intelligent advice' was ranked lowest by the architects (cf. Table 4). These two categories both assist architects in their routine knowledge processing tasks by automatically retrieving, and manipulating information, offering suggestions, etc. Particularly because architects appear to have little time or interest in documenting architectural knowledge, we had expected that all pro-active support 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. This 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, 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 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 are 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 [29]. A main characteristic of some of these platforms is that they follow a hybrid strategy for architectural knowledge sharing [12]. As Lago et al. argue, such a strategy "support[s] the knowledge producers in efficiently documenting the knowledge and the consumers in using it" [28], which would resolve the imbalance of producing and consuming knowledge we observed during our study.

5.1 Threats to Validity

We list possible limitations to our study by discussing the internal validity, construct validity and external validity following [22] and [31]. 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 (cf. [11]) leaves little room for selection bias or confounding variables.

To conform to *construct validity* we constructed our theory (cf. Figure 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.

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.

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.

Acknowledgment

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. We thank Philippe Kruchten for providing valuable suggestions and comments.

References

[1] M. Ali Babar and I. 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, P. Grisham, and D. Perry. Architectural Knowledge and Rationale - Issues, Trends, Challenges. *ACM SIGSOFT Software Engineering Notes*, 32(4):41–46, 2007.

[3] J. Bosch. Software Architecture: The Next Step. In *1st European Workshop on Software Architectures (EWSA)*, volume 3074 Springer LNCS, pages 194–199, 2004.

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

[5] R. Capilla, F. Nava1, and J. 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] P. Clements, R. Kazman, M. Klein, D. Devesh, S. Reddy, and P. Verma. The Duties, Skills, and Knowledge of Software Architects. In *6th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 2007.

[7] V. Clerc, P. Lago, and H. van Vliet. The Architect's Mindset. In *3rd International Conference on the Quality of Software-Architectures (QoSA)*, pages 231–249, Boston, USA, 2007.

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

[9] W. Fan, L. Wallace, S. Rich, and Z. Zhang. Tapping the Power of Text Mining. *Communications of the ACM*, 49(9):77–82, 2006.

[10] R. Farenhorst and R. C. de Boer. Knowledge Management in Software Architecture: State of the Art. In M. Ali Babar, T. Dingsøyr, P. Lago, and H. van Vliet, editors, *Software Architecture Knowledge Management: Theory and Practice*. Springer, 2009.

[11] R. Farenhorst, J. F. Hoorn, P. Lago, and H. van Vliet. What Architects Do and What They Need to Share Knowledge. Technical Report IR-IMSE-003, VU University Amsterdam, April 2009.

[12] R. Farenhorst, R. Izaks, P. Lago, and H. van Vliet. A Just-In-Time Architectural Knowledge Sharing Portal. In *7th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 125–134, 2008.

[13] R. Farenhorst, P. Lago, and H. van Vliet. EAGLE: Effective Tool Support for Sharing Architectural Knowledge. *International Journal of Cooperative Information Systems*, 16(3&4):413–437, 2007.

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

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

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

[17] J. F. Hoorn, M. E. Breuker, and E. 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.

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

[19] A. Jansen and J. 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.

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

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

[22] B. A. Kitchenham, S. L. Pfleeger, D. C. Hoaglin, and J. Rosenberg. Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE transactions on Software Engineering*, 28(8), 2002.

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

[24] P. Kruchten. What do Software Architects Really do? *The Journal of Systems and Software*, 81:2413–2416, 2008.

[25] P. Kruchten, R. Capilla, and J. C. Dueñnas. The Decision View's Role in Software Architecture Practice. *IEEE Software*, 26(2):36–42, 2009.

[26] P. Kruchten, P. Lago, and H. van Vliet. Building up and Reasoning about Architectural Knowledge. In *2nd International Conference on the Quality of Software Architectures (QoSA)*, volume 4214 Springer LNCS, pages 39–47, 2006.

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

[28] P. Lago, P. Avgeriou, R. Capilla, and P. Kruchten. Wishes and Boundaries for a Software Architecture Knowledge Community. In *7th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 271 – 274, 2008.

[29] P. Liang and P. Avgeriou. Tools and Technologies for Architectural Knowledge Management. In M. Ali Babar, T. Dingsøyr, P. Lago, and H. v. Vliet, editors, *Software Architecture Knowledge Management: Theory and Practice*. Springer, 2009.

[30] P. Liang, A. Jansen, and P. Avgeriou. Collaborative Software Architecting through Knowledge Sharing. In I. Mistrik, editor, *Collaborative Software Engineering*. Springer, 2009.

[31] D. Perry, A. A. Porter, and L. V. G. Empirical studies of software engineering: a roadmap. In *Conference on The Future of Software Engineering*, pages 345–355, Limerick, Ireland, 2000.

[32] J. Tyree and A. Akerman. Architecture Decisions: Demystifying Architecture. *IEEE Software*, 22(2):19–27, 2005.