Understanding How to Support Architects in Sharing Knowledge

Rik Farenhorst, Hans van Vliet
Department of Computer Science
VU University, Amsterdam, The Netherlands
{rik,hans}@cs.vu.nl

Abstract

In this paper we report on four years of case study research in a large software development organization, during which we built up understanding on how to effectively support software architects in sharing knowledge. By following a typical action research cycle we first diagnosed the architecting process of this organization, after which we designed and executed a therapy for the identified problems. The insights gained over the past few years have resulted in four lessons learned and have culminated in a theoretical framework of what architects do and what they need.

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. Over the past few years, more and 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 [6]. In addition, researchers have proposed various tools to support the architect, who is characterized as an all-round knowledge worker.

Perceiving architects as knowledge workers is one of the main reasons for the increased focus in academia on architectural knowledge management. The SHARK workshop series [3,20,21] has brought together researchers and practitioners to discuss methods, tools, and techniques to manage and share architectural knowledge. The obvious advances in this architectural knowledge community notwithstanding, we argue that some scoping and maturation would be beneficial to both practitioners and researchers. This ensures that the methods, tools and techniques developed actually help in solving real-world problems, and that a common frame of reference is created. The only way to do this, however, is by studying what architects in practice do and what kind of support they need.

Over the past four years we have studied the architecting process of a large software development organization, and focused on how to support architects effectively in sharing architectural knowledge. We conducted four consecutive case studies at this organization, which were all part of a typical action research cycle. In this paper we report on our gained understanding, which is reflected in a number of lessons learned. These lessons indicate that architects, do more than just architecting, that they need some incentives for sharing architectural knowledge, and that the best way to create these incentives lies in hybrid, just-in-time tool support. Based on the lessons learned and all related observations during the four case studies, we have established a descriptive framework consisting of architecting activities and support mechanisms for sharing knowledge. This framework helps researchers and practitioners to understand what architects really do and what kind of support for sharing architectural knowledge they need.

The remainder of this paper is organized as follows. In Section 2 we summarize our main lessons learned with respect to architectural knowledge sharing. In Section 3 we report on our case study research conducted over the past four years. In Section 4 we present a descriptive framework about what architects do and what kind of support they need. Section 5 concludes this paper.

2 Architectural Knowledge Sharing in Practice: Lessons Learned

Four years of case study research has significantly furthered our understanding with respect to architectural knowledge sharing in practice. We now understand better what architects do, what their responsibilities are, in which context they operate, and what role knowledge management plays or could play in their daily work. Our gathered understanding has culminated in the following lessons learned:
1. **Architects do more than just architecting.** Although creating architectural solutions is one of their primary responsibilities, architects spend a considerable amount of time on associated tasks and activities as well. Consequently, mechanisms to support architects in sharing knowledge should not only include architecting support such as modeling design decisions, or offering templates for architecture descriptions, but also include assistance for stakeholder communication, quality monitoring and searching relevant information.

2. **Architects do not share knowledge automatically.** Time is precious, especially for architects. By acting as ‘bridge’ between business and IT stakeholders, they spend much time on stakeholder communication, requirement negotiation, and on developing architectural solutions. Sharing architectural knowledge of and by themselves is, however, not an explicit part of their job description, especially not after the fact. Expecting architects to (extensively) share knowledge automatically is therefore not realistic. They need to be motivated to do so. To create these incentives, architectural knowledge sharing support should focus on assisting architects during architecting activities, instead of only offering repositories or templates to store their expertise and experiences after the fact. This reduces the overhead created by knowledge sharing and helps in convincing architects of the immediate and direct benefits related to this.

3. **Software architects benefit from lightweight, just-in-time, tool support.** Tools, methods or techniques to support architects in sharing knowledge should definitely be lightweight to better adapt to the various technical and non-technical tasks architects work on. This means that support mechanisms should be sufficiently user-friendly, usable, intuitive, and responsive, in order to motivate architects to keep using them. In addition, to properly accommodate architects in their knowledge needs, architectural knowledge should either be delivered to them or become accessible to them Just-in-Time. To realize this, architects may benefit from intelligent support during architecting activities or automated support for enrichment of codified architectural knowledge.

4. **Effective architectural knowledge sharing follows a hybrid knowledge strategy.** Codification of architectural knowledge using templates, repositories or metamodels, only supports architects to a certain extent. A substantial amount of the knowledge of architects will always remain tacit – i.e. impossible to articulate, let alone codify — and in these circumstances architects must be able to find the right person, instead of the right document. Software architecture knowledge management therefore should follow a hybrid strategy, incorporating both codification and personalization techniques.

3 **Four Case Studies: Action Research Cycle**

The lessons learned of the previous section summarize our understanding of architectural knowledge sharing in practice, by indicating what architects do (lessons 1 and 2) and what they need (lessons 3 and 4). This understanding is, however, not something that we have gained overnight. Instead, it was built up iteratively in the last four years (2005-2008), in which we conducted four consecutive case studies in CAP, a large Dutch software development organization. During the case studies we focused on several large architecture projects; we interviewed various architects, studied available documentation and examined methods, and tools used by the architects during these projects.

Each individual case study lasted around 5 months, and had its own scope, research questions, and methodology. The main results of the case studies have been published in earlier work [8–12]. Nonetheless, it is also worthwhile to look at the relationships between these case studies. A look at the ‘bigger picture’ shows that our qualitative research over the past few years closely resembles a typical action research cycle.

Action research is "grounded in practical action, aimed at solving an immediate problem situation while carefully informing theory [4]". In its most simple form there is a diagnostic and therapeutic stage, but the most prevalent action research description details a five phase, cyclical process consisting of [4]:

1. **Diagnosing.** Identification of the primary problems that are the underlying causes of the organizations desire for change. In this phase certain theoretical assumptions about the nature of the organization and its problem domain are developed.

2. **Action planning.** Specification of organizational actions that should relieve or improve the primary problems. In this phase a theory is formulated about how to address the earlier identified problems.

3. **Action taking.** Researchers and practitioners collaborate in the active intervention into the client organization, causing certain changes to be made.

4. **Evaluating.** Researchers and practitioners evaluate the outcomes of the action taken. This evaluation includes determining whether the theoretical effects of the action were realized, and whether these effects relieved the identified problems.
5. **Specifying learning.** The restructuring of organizational norms or theory to reflect the new knowledge gained by the organization during the research. This phase may also indicate further action research interventions that are worth diagnosing. While the activity of specifying learning is formally undertaken last, this reflection phase is usually an ongoing process that plays a role throughout all other four phases.

The case studies we conducted in the last four years clearly resemble the typical phases of an action research cycle. A schematic overview of this cycle is depicted in Figure 1. In the following subsection we will elaborate upon each case study in detail. For each study we explain to which action research cycle it corresponds best, and by summarizing the results of each study we show how our understanding of how to support architects in sharing architectural knowledge was built up step-by-step.

### 3.1 Case study 1: Diagnosing the Architecting Process

In the first case study, which was conducted early 2006, we explored the status quo in CAP with respect to architectural knowledge sharing [8, 11]. Therefore, it corresponds well to a diagnosis phase of an action research cycle, in which the main problems and points for improvement are identified. In this case study we mainly focused on the architect’s activities and his roles and responsibilities in the architecting process.

We quickly found that most mechanisms in place to share knowledge were rather ineffective. For example an existing system for storing architecture guidelines and best practices was barely used because architects deemed its added value rather low. In order to understand what caused the problems with these tools and to indicate some points for improvement, we needed to better understand how the architecting process looked like. Our main insights during this case study were that architects do more than just creating architectural solutions and that incentives needed to be created to stimulate architectural knowledge sharing.

While studying the architecting process of CAP, our first discovery was that architects do much more than making design decisions or creating architectural descriptions. We found that architects are also responsible for communication with internal and external stakeholders, which showed itself through workshops, coffee room chats, requirements negotiations, etc. In addition, many projects require some quality monitoring of architects, during which they review existing architectural documentation, verify whether architectural guidelines or rules are adhered to, check whether an architecture solution conforms to reference architectures of customer organizations, and during which they assess the feasibility of the architectural solution.

The insights about these activities and responsibilities of architects are confirmed in recent literature on software architecting. Kruchten states that software architecting involves a lot of consensus decision making in which architects balance between quality criteria, stakeholder concerns, and requirements [19]. Clements et al. argue that “[the architect’s] job is far more complex than just making technical decisions, although clearly that remains their most essential single duty” [6]. They found in a large-scale study in practice that architects frequently interact with stakeholders, are involved in organization and business related issues, but primarily guide the architecting process.

The second main discovery during this case study was an apparent lack of motivation of architects with respect to proactively sharing architectural knowledge. It became clear that part of the reason for this was the fact that available knowledge management tools had a poor usability, steep learning curve and quickly outdated contents. Important to notice, however, is that architects also did not seem to have – or make – time to give these tools a fair chance.

Lack of time, or trust in a tool for storing or sharing knowledge are well-known issues in knowledge management literature. To deal with these issues, incentives for sharing architectural knowledge should be created [13]. One way to do this is by making codification of architectural knowledge easier. Although some knowledge is inherently tacit in nature and therefore impossible to codify, i.e. detach from its owner [23] – researchers do stress that a group’s or community’s performance increases significantly if everyone is informed of each other’s expertise or when more explicit knowledge is available to internalize [7].

Our diagnosis at CAP also indicated that some improvements can be made with respect to community building. We found that little explicit communication lines were available between architects and other important stakeholders, such
as developers, project managers and maintainers. Consequently, little social ties were established. Based on organizational learning theory we argued that improving the status quo at CAP involves creating trust and improving communication mechanisms in order to increase the architects’ motivation for cooperation and sharing knowledge [2].

3.2 Case study 2: Planning Architectural Knowledge Sharing Support

The second case study at CAP was conducted early 2007 and focused on establishing properties that characterize effective architectural knowledge sharing tool support [10]. This case study makes for a typical action planning phase of an action research cycle, because in this study we formulated a theory to address the diagnosed problems of the previous case study.

We started to look into knowledge management and software architecture literature to see what kind of dos and don’ts exist with respect to architectural knowledge sharing support. In parallel we evaluated existing tools in CAP to identify strengths and weaknesses. We culminated all insights into a number of desired properties for architectural knowledge sharing tools.

One of the main insights gained in this case study is that architects benefit from a so-called ‘hybrid’ strategy in sharing architectural knowledge. This means that a proper combination between a codification and personalization strategy needs to be found [14]. Codification seems to the prevalent strategy in research in practice to manage and share this knowledge, according to a preliminary survey [1], but it is also argued that there are good reasons to choose for personalization mechanisms as well, or go for a hybrid approach. This is acknowledged by Huysman and Wulf, who found that when sharing experience, people prefer to look for support from personal networks rather than from electronic networks to gain knowledge about the knowledge [17]. Huysman also warns for falling in an ‘ICT-trap’, which is caused by an underlying, but unrealistic, assumption that ICT can always support and improve knowledge sharing within organizations [16].

Other desired properties for architectural knowledge sharing tools we identified include support for stakeholder-specific content, easy manipulation of content and collaboration. The need for collaboration, in combination with hybrid knowledge sharing is acknowledged by Brink, who defines three categories that form technological enablers for knowledge sharing [24]: a knowledge repository (for sharing explicit knowledge); a knowledge routemap (for sharing explicit and tacit knowledge), and a collaborative platform (for sharing tacit knowledge).

We used the above insights to design and implement a prototype web portal that adheres to all desired properties for sharing architectural knowledge. This part of the case study corresponds well to the description of an action plan in which the formulated theory is made concrete and executable. The ‘action taking’ phase itself, in which we experimented with the portal at CAP’s architecting process, was part of the next case study.

3.3 Case Study 3: Supporting Architects Just-In-Time

The third case study at CAP was conducted mid-2007 and is described in more detail in [9]. In this case study we experimented with a portal to improve the status quo at CAP with respect to architectural knowledge sharing. During our experimentation we evaluated with a number of architects what they liked and disliked. This case study is therefore a combination of the ‘action taking’ and ‘evaluation’ phases of action research.

Before we started experimenting with our portal we examined in more detail existing tools for architectural knowledge sharing at CAP. We elicited from the architects what the limitations are of these existing tools and which requirements they have for a potentially new environment. This requirement elicitation was a continuation of the diagnosis we conducted during our first case study (cf. Section 3.1), but also a validation of the theory about effective tool support we established in our second case study (cf. Section 3.2).

The requirements elicitation part of this case study showed that architects need effective support for searching relevant information, managing documentation, and assistance to easily get in touch with colleagues or other stakeholders. When we aggregated these insights, we found that CAP’s architects were apparently in need of ‘Just-in-Time’ architectural knowledge, which we defined as access to and delivery of the right architectural knowledge, to the right person, at any given point in time. Increasing the access to and delivery of architectural knowledge will boost reuse of architectural knowledge, stimulates learning among practitioners, and assists in assessing quality (e.g. by pointing architects to reference architectures, best practice databases, and quality frameworks).

The importance of Just-in-Time knowledge is acknowledged by Kerschberg and Jeong, who argue that effective decision making demands that the decision makers are able to efficiently search through the mountains of data to find the right knowledge pieces at the right time [18]. Since architects are decision makers [5], who keep a backlog of smaller needs, issues, problems and challenges [15], they definitely benefit from access to and delivery of relevant architectural knowledge at the right point in time to make well-founded decisions.

Experimentation with our web portal indicated that architects perceive it as a definite improvement over the exist-
ing tools at CAP. The fact that several community building – or personalization – mechanisms (e.g. yellow pages, discussion boards) were combined with codification techniques (e.g. repositories, project environments) offered the architects an allround platform to share architectural knowledge in various ways. The intuitive user interface further helped motivate architects to actually use these tools, i.e. create the necessary incentives for sharing architectural knowledge.

Apart from the obvious strengths of our portal, during our evaluation the architects also indicated a number of possible improvements. Interestingly, the architects did not ask for new kind of functionality or features, but instead wished for more elaborate support for some of the core parts of our portal, including more focus on collaboration, communication and integration with other tools. We decided to take this feedback seriously and planned for another case study of action taking and evaluation to address these requirements.

3.4 Case Study 4: Supporting Architects using Wikis

The fourth case study at CAP was conducted in the first half of 2008 and is described in more detail in [12]. In this case study we explored the successfulness of using wikis for sharing architectural knowledge. Wikis are strong in community building and in supporting collaboration, and during the previous case study we found that architects needed more extensive support in these areas. In terms of placing this case study in an action research cycle it is similar to the previous one; the experimentation and evaluation of a tool environment corresponds well to the ‘action taking’ and ‘evaluation’ phases of action research.

In this case study we learned that wikis have several characteristics which make them very suitable as knowledge sharing environment in the architecting process. Facilities to hold discussions, group editing features and the very intuitive user-interface make wikis a very lightweight yet versatile platform. In addition, the commercial enterprise wiki we experimented with offers various integration mechanisms to other tools (such as Sharepoint websites) and CAP’s file shares, which was greatly appreciated by the architects and was considered a clear improvement over the portal we had shown to them in the previous case study. They now finally had a single point of entry to a web of knowledge at their disposal, which easily combines management of very specific architectural knowledge (e.g. design decisions, best practices, patterns), but also related knowledge crucial for any project (project documentation, progress reports, handbooks).

Wikis are also quite suitable in creating a ‘community of architects’ in which everybody knows from each other where certain expertise, or experience resides. This helps architects in various activities such as conducting reviews, writing reports, or communication with stakeholders. The enterprise wiki we experimented with offers various plugins which add various interesting features, such as modeling support, generation of statistics and overviews, and even text mining features in combination with database integration. We argue that these aspects can further alleviate architects in their daily work by offering ‘smart’ or pro-active support when needed, in a non-intrusive way (e.g. automatically identifying patterns or rules from raw wiki content).

In this case study we found that wikis are able to create incentives for architectural knowledge sharing, because in a short amount of time a lot of architectural knowledge was contributed by the architects who participate in a pilot. We did learn, however, that certain so-called ‘wiki patterns’ need to be kept in mind to prevent things going out of hand [22]. Examples are the appointment of wiki ‘gardeners’ who keep the wiki organized, or the establishment of templates to enable uniform codification of specific types of knowledge. These insights showed that a versatile tool in itself is not enough; process rules and guidelines are needed to make effective use of the tool and to position it correctly in the architecting process.

3.5 Summary

The previous subsections show how our research in CAP over the past four years has furthered our understanding with respect to architectural knowledge sharing in practice. The insights gained during the four consecutive case studies relate to the four overarching lessons learned listed in Section 2 as follows:

- During our diagnosis in the first case study we found that architects do more than just architecting (lesson 1) and that incentives need to be created to motivate architects to share architectural knowledge in the first place (lesson 2).
- Our action planning in the second case study led us to belief that a hybrid strategy for sharing architectural knowledge is the most effective approach (lesson 4).
- During the third case study our analysis of the requirements from the architects indicated that architects are in need of Just-In-Time architectural knowledge (lesson 3), but we also found that our web portal fell a bit short in terms of collaboration and communication support. The fact that architects mentioned these shortfalls reconfirmed our belief that architects communicate a lot and do much more than just creating architectural solutions (lesson 1).
- Our last case study showed us that a wiki environment is a more suitable lightweight platform for sharing architectural knowledge than our initial web portal (les-
son 3), while offering a balanced mix between codification and personalization mechanisms (lesson 4). Our experimentation also showed that a wiki does create the required incentives for knowledge sharing if positioned correctly in the architecting process, because it offers various mechanisms to assist architects during their primary activities (lesson 2).

4 What Architects Do and What They Need

After four years of case study research we claim to now have a decent understanding of how to support architects in sharing knowledge. Apart from identifying a number of lessons learned in Section 2 we have culminated our insights of all four consecutive case studies in a descriptive framework, which corresponds to a ‘specifying learning’ phase of an action research cycle, as depicted in Figure 1.

This framework focuses on what architects in practice do and what they need. It consists of two parts: a) a characterization of what architects really do, captured in a number of typical architecting activities, and b) a theory on what architects need with respect to sharing architectural knowledge, captured in a number of support mechanisms. The complete framework is depicted in Table 1.

Although the limited size of this paper prohibits us from elaborating upon each subcategory in detail, the categories of our framework are clearly rooted in the insights gathered during our four case studies. Likewise, an apparent link between our lessons learned and the various categories and subcategories can be quickly discovered. In the remainder of this section, we illustrate this by a number of examples.

The first part of our framework consists of six categories of architecting activities. From these six categories it can be immediately observed that architects do much more than just architecting. Of course taking architectural decisions is an important responsibility of architects, but in order to take well-founded decisions, a lot of communication and collaboration is involved as well. Communicating with colleagues involves informing and talking to them about various architectural issues, but also delegating tasks to them or steering them when required. In contrast to this more ‘internal’ communication, communication with stakeholders focuses especially on customer relationships, selling solutions and explaining or convincing solutions to (non-technical) stakeholders. This activity thus includes organizational politics and related techniques that are usually not necessary when communicating with colleagues in the team. Storing architectural knowledge includes all activities related to express application-generic or application-specific knowledge in a certain form, such as a document, repository, model or template, the primary driver being reuse of this knowledge. To expand knowledge means increasing one’s knowledge or staying up-to-date on architecture-related topics using various mechanisms such as presenting, publishing, lecturing, reading, and retrieving knowledge using internal or external sources. Lastly, the control quality category includes all review and audit tasks architects have to ensure that the architecture design is adhered to, guidelines are followed and the stakeholders’ requirements are met.

When examining the six categories of architecting activities, it is not hard to see that during each of these activities, architects may benefit from support for sharing architectural knowledge. The second part of our framework depicts six main categories of such support. It can be observed from Table 1 that these categories combine support for codification of architectural knowledge (e.g. templates, repositories and modeling support) with personalization support that strive for community building. The Just-in-Time paradigm is well reflected in the ability to search for architectural knowledge in an effective and lightweight manner, but also by offering several mechanisms to enrich architectural knowledge or offer intelligent support during the various architecting architects are involved in. Last but not least is support for managing architectural decisions and related architectural knowledge, since this assists architects in one of the primary tasks they have in a typical software development project. Our experimentations with both a web portal and a wiki environment has indicated that only if architectural knowledge sharing support is versatile and lightweight enough, then architects are motivated to actually make use of this support.

5 Conclusions

By conducting four years of case study research in a large software development organization we have built up a solid understanding of what architects in practice do and what kind of support they need with respect to architectural knowledge sharing. We discovered that architects do much more than taking design decisions or producing architectural documentation. Architecting requires that they spend a significant portion of their time on communication with colleagues or other stakeholders, quality control, and on expanding their knowledge in any way they can. Consequently, to assist architects in sharing knowledge during all these activities, support for codification of architectural knowledge is not sufficient. Explicit mechanisms for managing decisions, searching relevant information or finding the right colleague are at least equally important to ensure that architectural knowledge is acquired ‘Just-in-Time’. To further alleviate the workload of architects, intelligent support or enrichment of architectural knowledge may prove valuable as well. Our experiment with a web portal and a wiki indicated that only lightweight and versatile knowledge management platforms which include aforementioned support create the required incentives to motivate architects...
to share architectural knowledge.

By conducting four case studies we were not only able to diagnose the problems of a large Dutch software development organization with respect to sharing architectural knowledge, but also able to plan and execute actions to address these problems. The architectural knowledge management domain is by and large unchartered territory, but this type or qualitative research proved quite effective and has led to concrete knowledge sharing improvements in CAP. In addition, we argue that our framework could help both researchers and practitioners to understand what architects really do and what kind of support for sharing architectural knowledge would be most suitable in specific contexts.

Even though our descriptive framework is rooted in state-of-the-art literature in software architecture and knowledge management, its limitation is that it is based on practical findings of just one software development organization. In order to validate our framework on a much broader scale, we are currently conducting large-scale survey research in which about 250 architects from three large software development organizations in the Netherlands participate.

Acknowledgements

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.

References

### Table 1. Descriptive Framework: What Architects Do and What They Need

<table>
<thead>
<tr>
<th>What architects do</th>
<th>What architects need</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Take architectural decisions</strong></td>
<td><strong>Management of architectural decisions</strong></td>
</tr>
<tr>
<td>- Take architectural decisions</td>
<td>- Overview of important architectural decisions and the underlying rationale</td>
</tr>
<tr>
<td>- Weigh pros and cons (and tradeoffs) of arch. solutions</td>
<td>- Visualization of current problems or bottlenecks in a project</td>
</tr>
<tr>
<td>- Meet stakeholder demands and concerns</td>
<td>- Insight in the relationships between taken decisions</td>
</tr>
<tr>
<td>- Provide rationale by linking to guidelines, rules, etc.</td>
<td>- Templates for describing architectural decisions</td>
</tr>
<tr>
<td>- Reason about the impact of decisions on the architecture</td>
<td>- Traceability between stakeholder reqs. and decisions</td>
</tr>
<tr>
<td>- Build consensus among stakeholders</td>
<td>- Overview of conflicting stakeholder requirements</td>
</tr>
<tr>
<td>- Trace conflicts or other relationships between decisions</td>
<td>- Visualization of changes in decisions through time</td>
</tr>
<tr>
<td><strong>Communicate with colleagues</strong></td>
<td><strong>Codification of architectural knowledge</strong></td>
</tr>
<tr>
<td>- Inform colleagues about (the results) of architecting tasks and activities</td>
<td>- Modeling support for architectural solutions</td>
</tr>
<tr>
<td>- Talk to colleagues about architectural topics</td>
<td>- Central storage facilities for architectural best practices</td>
</tr>
<tr>
<td>- Delegate tasks or responsibilities to colleagues</td>
<td>- Templates for architectural deliverables</td>
</tr>
<tr>
<td>- Hold discussion among colleagues about architecture</td>
<td>- Central repository for architectural documentation</td>
</tr>
<tr>
<td>- Approach colleagues for help or a second-opinion</td>
<td>- Automated way to model traceability between documents</td>
</tr>
<tr>
<td>- Explain architectural principles and rules to colleagues</td>
<td>- Tool environment to describe architectural guidelines together with colleagues</td>
</tr>
<tr>
<td>- Motivate and/or steer colleagues</td>
<td><strong>Community building</strong></td>
</tr>
<tr>
<td><strong>Communicate with stakeholders</strong></td>
<td>- Insight in who is working on which things in the project</td>
</tr>
<tr>
<td>- Report about the status and progress to stakeholders</td>
<td>- Central system to hold discussions with colleagues or stakeholders</td>
</tr>
<tr>
<td>- Elicit stakeholder requirements, wishes and concerns</td>
<td>- Ability to signal colleagues about specific documentation</td>
</tr>
<tr>
<td>- Explain architectural decisions to stakeholders</td>
<td>- Central environment to collaborate with colleagues on producing deliverables or to peer-review each others work</td>
</tr>
<tr>
<td>- Share the vision behind the architecture with stakeholders</td>
<td>- Ability to chat with colleagues or other stakeholders</td>
</tr>
<tr>
<td>- Negotiate in stakeholder conflicts</td>
<td>- Retrieve expertise or experience from colleagues</td>
</tr>
<tr>
<td>- Promote communication between business and IT</td>
<td><strong>Intelligent support</strong></td>
</tr>
<tr>
<td>- Convince stakeholders of the contribution of arch. solutions</td>
<td>- Feedback while producing architectural documentation</td>
</tr>
<tr>
<td>- Sell architectural solutions to stakeholders</td>
<td>- Advice on which architectural decisions to take and why</td>
</tr>
<tr>
<td>- Learn from colleagues or stakeholders about arch. principles</td>
<td>- Suggestions and tips on which arch. guidelines to apply</td>
</tr>
<tr>
<td>- Study architectural decisions from completed projects</td>
<td>- Automatically generated documentation based on specific input parameters</td>
</tr>
<tr>
<td>- Unravel the reasoning behind or rationale of arch. decisions</td>
<td>- Automatic delivery of relevant architectural information</td>
</tr>
<tr>
<td>- Build up knowledge by studying arch. patterns and styles</td>
<td>- Notifications of the availability of new relevant publications</td>
</tr>
<tr>
<td>- Transfer knowledge during discussions with colleagues</td>
<td>- Notifications of newly available arch. documentation</td>
</tr>
<tr>
<td>- Search for relevant information on Internet or Intranet</td>
<td><strong>Enrichment of architectural knowledge</strong></td>
</tr>
<tr>
<td>- Attend lectures and presentations on relevant topics</td>
<td>- Automatic retrieval of arch. guidelines within documents</td>
</tr>
<tr>
<td>- Read relevant publications and articles on architecture</td>
<td>- Annotation support for architectural concepts in documents</td>
</tr>
<tr>
<td>- Search for relevant information on Internet or Intranet</td>
<td>- Detection of architectural decisions within deliverables</td>
</tr>
<tr>
<td>- Transfer knowledge during discussions with colleagues</td>
<td>- Reports about application and use of architectural rules and guidelines in projects</td>
</tr>
<tr>
<td>- Attend lectures and presentations on relevant topics</td>
<td>- Overview of open design issues</td>
</tr>
<tr>
<td>- Read relevant publications and articles on architecture</td>
<td>- Tracing of stakeholders’ contribution to a document</td>
</tr>
<tr>
<td>- Unravel the reasoning behind or rationale of arch. decisions</td>
<td>- Enrichment of documentation with metadata</td>
</tr>
</tbody>
</table>