Abstract

Practices for architectural knowledge management (AKM) may alleviate the challenges involved with GSD. We have conducted empirical research at a large Dutch IT service provider to validate a set of practices for architectural knowledge management in GSD and to specifically investigate the relation between the number of sites and the perceived usefulness of these practices. The results show that AKM practices supporting a personalization strategy towards knowledge management are perceived to be more useful than practices that support a codification strategy. Further, the usefulness of AKM practices in general is confirmed. Finally, we observe a peak in the perceived usefulness of AKM practices in projects that evolved to a multi-site situation. This high perceived usefulness denotes a more critical need to plan for AKM practices in advance.

1. Introduction

Within the field of software engineering, increasing attention is paid to Global Software Development (GSD). In GSD, software development takes place at geographically distributed development sites. Although GSD can result in benefits such as reduced development time and increased availability of skilled resources [7], global software development poses additional challenges as well. Overviews of these challenges have been widely reported [1, 5, 9, 10, 18] and include lack of informal contact, language differences, and coordination complexity [1].

A specific discipline within the field of software engineering is software architecture. Within the software architecture community, an increasing interest in architectural knowledge management is recognized [3, 20, 22, 28]. Architectural knowledge management (AKM) involves capturing and communicating the design decisions that lead to a software system, including underlying rationale and context [2].

Using architectural knowledge effectively may help in overcoming the challenges and issues encountered in GSD. However, we are currently lacking detailed insight into AKM practices that can effectively be applied in a GSD setting. Specifically, we are interested in identifying the relation between the perceived usefulness of AKM practices and the number of sites in a software development project because this will offer further guidance on applying the AKM practices.

To gain insight into the relation between perceived usefulness of AKM practices and the number of sites, we built upon previous work in which we identified relevant AKM practices based on literature and experience reports from the requirements engineering discipline [8]. In this research, we validate these AKM practices by conducting empirical research at a large IT service provider in the Netherlands.

Our research shows that when the number of sites involved in a software development project increases the perceived usefulness of AKM practices does not necessarily increase. The perceived usefulness of AKM practices has its maximum in software development projects with three sites. Further investigation revealed that not all projects with three sites are initially set up as such; usually, they start with two sites and evolve to their current situation with three sites. The high perceived usefulness for some AKM practices at three sites denotes a need to have these practices implemented proactively.

Furthermore, the usefulness of AKM practices in general is confirmed. Yet, some well-known global practices, such as having a clear project structure with communication responsibilities assigned or having a single repository for architecture artifacts, are not perceived as being as useful as one may expect. Finally, our research shows that practices that support a personalization strategy towards knowledge management are perceived as more useful than practices that support a codification strategy towards knowledge management.

This paper is structured as follows. Section 2 provides an overview of related work in the field of software architecture, architectural knowledge management, and global
software development. Section 3 lists the research question posed for this research. In Sect. 4 we describe the approach used for the research. The analysis and the results are described in Sect. 5 and discussed and validated in Sect. 6. Finally, Sect. 7 lists our conclusions and directions for future work.

2. Related Work

The requirements engineering discipline is one of the first software engineering disciplines in which research has been conducted to identify practices that alleviate the challenges that are involved with GSD. The work of Damian et al. [11, 12, 13] lists requirements engineering challenges and their (proposed and experienced) solutions. Bhat et al. [6] provide insight into the challenges with requirements engineering in off-shore outsourced situations and lists ways to overcome these challenges.

A resemblance between a set of requirements for a software system and a set of design decisions taken for that software system has been noted [27]. Avgeriou et al. [2] report on the identification of architectural decisions that pertain to the problem space instead of the solution space. The problem space is populated with requirements. Consequently, the challenges experienced in the requirements engineering discipline in GSD may have their counterpart in the discipline of architectural knowledge management in GSD, as shown by [9, 10]. In our research, we combine this topic with other challenges in knowledge sharing in a GSD situation as observed by [4, 14].

In previous work, we gave an overview of the architectural knowledge management practices that can be applied in a GSD setting [8]. We defined a light-weight pattern language for describing the practices which helps us in performing subsequent analyses. Our current research is aimed at validating these AKM practices in practice.

Fahrenhorst et al. [15, 16] list important prerequisites for architectural knowledge sharing and how these prerequisites can be supported by effective tool support. Especially within the collaboration-intensive GSD, these prerequisites are important. In [3], Babar et al. further focus on different strategies for sharing and managing architectural knowledge, following knowledge management literature: personalization and codification. A personalization strategy promotes interaction between knowledge workers; knowledge is kept by its creator. A codification strategy supports codifying knowledge and storing it in a repository, structured or unstructured. Babar et al. conclude that a hybrid approach between personalization and codification is needed most.

In a workshop on sharing and reusing architectural knowledge [2], insight was obtained into the dos and don’ts with respect to architectural knowledge. The workshop participants included both academia and industry. The workshop results show that the majority of the dos pertain to personalization, but that both knowledge management strategies need to be supported by the service provider’s culture. In addition, the workshop reported other important dos, such as assigning a traveling architect, establishing peers, and allowing for periodic after-the-fact reflection.

3. Research Question

For gaining further insight into the actual AKM practices for GSD, we are interested in determining the relation between the number of development sites involved in a global software development project and the perceived usefulness of AKM practices. This will further help us to build a set of AKM practices with additional guidance on their application. To this end, we formulate our research question as follows:

How does the number of sites involved in a software development project influence the perceived usefulness of AKM practices for that project?

4. Research Approach

4.1. Organization Context

We conducted our research at the Dutch branch of a large, international IT service provider. The Dutch branch employs over 6,000 employees spread across a number of business sectors, e.g. public, finance, and energy.

The service provider aims for standardization of its software development processes1 and associated tools. Associated tools include a central configuration management system, standardized electronic project workplaces, and standardized collaboration and communication tooling.

The software development methodology in use by the service provider can be characterized as “mixed delivery”. Accordingly, the service provider chooses for the appropriate proportion of activities performed on-site at the customer and works towards an ideal mix between on-site, off-site (at one of the service provider’s offices), near- and off-shored software development.

4.2. Approach

We conducted seven semi-structured interviews at prominent multi-site projects at the IT service provider to collect an initial view of the software development practices that are in place at the service provider. The interviews focused on the context of the software development project, the size of the software development project (both

1The organization aims for CMMI level three [24].
Table 1. Questions relevant to the research

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>8. Please list all locations where project members were performing software development activities.</td>
<td></td>
</tr>
<tr>
<td>10. What practices for architectural knowledge management can be useful within your project?</td>
<td>(Yes, Neutral or Don’t know, No)</td>
</tr>
<tr>
<td>Frequent interaction between the teams involved</td>
<td></td>
</tr>
<tr>
<td>Representatives of each team visit other teams</td>
<td></td>
</tr>
<tr>
<td>Face-to-face project kick-off meeting</td>
<td></td>
</tr>
<tr>
<td>Email, mailing list or telephone to quickly get information</td>
<td></td>
</tr>
<tr>
<td>Have a clear project structure with clear communication responsibilities</td>
<td></td>
</tr>
<tr>
<td>Have a single repository for architecture artifacts</td>
<td></td>
</tr>
<tr>
<td>Traveling to other project locations</td>
<td></td>
</tr>
<tr>
<td>Knowing who is who across the project (directory)</td>
<td></td>
</tr>
<tr>
<td>Having a shared infrastructure for work products and source code (configuration management)</td>
<td></td>
</tr>
</tbody>
</table>

5. Analysis and Results

5.1. Demographic Information

The total population consisted of 363 employees of the IT service provider. These employees, all members of the architecture mailing list, are performing architecting or architecting-related activities. We received 132 responses, which corresponds to a response rate of 36.4%. Not all of these 132 responses were complete. We selected only the responses that provided answers to the questions that are relevant to our research (see Table 1). This resulted in a set of 114 responses, corresponding to an adjusted response rate of 31.4%. We used these responses as a basis for our research. Although the data does not permit to determine the total number of project due to confidentiality purposes, analysis of the responses reveals that the responses include at least 70 different software development projects.

We asked the participants to complete the questionnaire with a certain (past or ongoing) software development project in mind. Consequently, by indicating the perceived usefulness of a specific AKM practice, a respondent does this for the software development project at hand.

Table 1 lists the parts of our questionnaire relevant to this paper. We inquired a list of all the locations where software development occurs for possible future geographic analysis. Furthermore, we decided to use a three-point Likert-scale [19, 23] with values “Yes”, “Neutral or Don’t Know”, and “No”. Specifically, we grouped “Neutral” and “Don’t Know” because we are primarily interested in the significant outlying results, forcing the respondents to a clear statement.

5.2. Data Preparation

Table 2 shows the distribution of the 114 respondents over the number of sites in the software project they were involved in. A site is defined as a designated location, possibly the customer’s location, where personnel of the IT service provider is stationed.
Table 2. Number of responses per number of sites

<table>
<thead>
<tr>
<th>No. sites</th>
<th>No. respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

We regard the number of respondents in each individual category of 4, 5, 6, and 7 sites too low to draw conclusions from. Consequently, we sum up the individual responses of these categories, and form a new group (“4–7 sites”), with 17 responses.

For the analysis of the usefulness of a specific AKM practice, we only took into account respondents who rated the perceived usefulness of that practice by answering the appropriate subquestion of question 10. Hence, the response rate per AKM practice may differ slightly because not all participants answered all subquestions of question 10.

5.3. Perceived Usefulness of AKM Practices

Based on our research questions we relate the value contributed to each AKM practice to the number of sites involved in the software development project of that respondent. We describe our results in the remainder of this section.

We list each practice and start by summarizing the intention of the practice from [8]. Next, we summarize the results of the perceived usefulness of the practice by the respondents. The Appendix shows the responses for each AKM practice graphically.

1. Frequent interaction between the teams involved

This practice intends to let practitioners from different sites interact frequently with each other. Interaction may be done through a variety of means, e.g. by using collaboration software such as video-conferencing or wikis, or through on-site, face-to-face interaction. This practice is primarily important for architects to share their views on the architecture with other project stakeholders.

Other research further distinguishes between e.g. audio support and video support [26]. Based on the standardization in the IT service provider’s software development methodology and the combined use of audio support and video support, we have not made this distinction.

We observe a positive attitude towards the usefulness of this practice in GSD. Regardless the number of sites, more than 50% of the respondents regards frequent interaction between the teams involved to be valuable.

2. Representatives of each team visit other teams

This practice intends to obtain better integration between teams involved in the software development project by having teams visit each other (e.g. during joint architecture meetings or pair software design or development).

The respondents value this practice, although not as much as some other practices. Furthermore, we observe a clear (positive) peak in the perceived usefulness of this practice at projects that involve three sites, and a lowered value from respondents involved in four or more sites.

3. Face-to-face project kickoff meeting

This practice intends to establish initial relationships across sites by bringing together all project members to hold a joint kickoff meeting.

The respondents clearly value this practice: regardless of the number of sites involved, more than 70% of the respondents value this practice; its perceived value shows a slight increase with the number of sites increasing, but a peculiar drop at three sites.

4. Email, mailing list or telephone to quickly get information

This practice intends to quickly collect information on a given topic of interest. The service provider in our research has implemented a group mailing address that includes all Dutch architects.

The results show that our respondents clearly value this practice; again, more than 70% of the respondents perceive this practice as being useful, regardless of the number of sites. The pattern, however, is different. We observe a drop in perceived usefulness at two sites, and a steady increase towards three and four or more sites.

5. Have a clear project structure with clear communication responsibilities

This practice intends to maintain open communication lines between well-defined stakeholder roles.

We observe that the respondents value this practice less than the previously mentioned practices. We would have expected to observe an increase in perceived usefulness since projects with a higher number of sites often include more project members and a larger variety of roles involved. Instead, we observe that only 30.8% of the respondents who are involved in a project with four sites or more value this practice.

6. Traveling to other project locations

This practice intends to obtain better integration between teams involved in the software development project by hav-
ing teams from different development sites visit each other physically.

About 40 % of the respondents value this practice. One exception exists: 66.7 % of the respondents involved in a project with three sites value the practice.

7. Knowing who is who across the project (directory)

This practice intends to allow project members to quickly know who is working within the project. This can be done by providing a directory, consisting of e.g. a mugshot and phone or email contact information.

We observe no specific impact of the number of sites involved onto the perceived usefulness of this AKM practice. This is reflected by a correlation coefficient of -0.041. The perceived usefulness of this practice is not as high as we had expected from previous research; in [9] we showed that the yellow pages containing directory information were the most popular of all system-related information.

8. Have a single repository for architecture artifacts

This practice intends to build a repository to store architectural decisions including the rationale of these decisions.

The respondents value the perceived usefulness of this practice lowest of all AKM practice; at most 40 % of the respondents at each site perceive this practice as useful. In addition, we observe a drop in perceived usefulness at the group of respondents working in software development projects that include 4 or more sites. This is reflected by a moderately low correlation of -0.114.

9. Have a shared infrastructure for work products and source code (configuration management)

This practice intends to have a shared environment or infrastructure where project members can share work products like documents, plans, and source code.

This AKM practice is in general perceived as useful, regardless of the number of sites involved. A prime reason for this may be that this practice is highly supported by the IT service provider’s software development methodology, as described in Sect. 4.1.

6. Discussion of the Results

6.1. Analysis of the Number of Sites

So far we have presented the perceived usefulness of each AKM practice related to the number of development sites. Now we can combine these results to explain differences observed. Figure 1 shows these combined results. We make a number of observations based on these results:

- First, the perception of the usefulness of the AKM practices does not differ between projects with one development site (i.e local) and projects that involve two sites (i.e. distributed). Only the AKM practices “using email, mailing list or telephone to quickly get information” and “have a clear project structure” show a drop of more than 10 % in perceived usefulness at two sites. The difference in perceived usefulness of the remaining seven practices does not differ significantly between one or two sites involved.

- Secondly, we observe a peak in perceived usefulness at software development projects that involve three sites. Only the AKM practice “frequent interaction” shows a drop; all other practices show an increase in perceived usefulness. Five out of nine AKM practices show an increase of more than 10 %.

We have statistically analyzed the observed differences as listed above to verify whether the difference in perceived usefulness observed at software development projects with three sites is significant. We assume that the population follows a normal distribution and applied the chi-squared test [17] to test whether the observed variance in the results is statistically significant. To this end, we formulated the following hypotheses:

- $H_0$ – The responses of the group with three sites does not differ significantly from each of the other groups.

- $H_1$ – The responses of the group with three sites differ significantly from each of the other groups.

For each AKM practice, we compared the responses of each group (one site, two sites, three sites, and four or more sites) with the other groups. In general, we observe very high $p$-values (above 0.5) which result in not being able to discard the null hypotheses; the observed difference in perceived usefulness of a practice per number of sites involved does not appear to be significant. However, we do observe significantly lower $p$-values (about 0.1) when we observe the usefulness of AKM practices focusing on visiting other teams and traveling to other project locations of three sites. Although the $p$-value observed is still above the 0.05 threshold that is common in empirical research, these lower $p$-values do urge us to investigate this more thoroughly (see Sect. 6.3). The main reason for not having obtained statistically significant result is that the data is highly skewed (the number of responses across the different number of sites varies, cf. Table 2).

6.2. Analysis of AKM Practices

As with the difference in perceived usefulness of practices depending on the number of sites involved, we combined the results to make some observations. First of all, a

---

3For this reason, we have chosen not to list all $p$-values.
difference exists between the perceived usefulness of codification practices and personalization practices. In [8], we described the codification practice “Have a single repository for architecture artifacts” and this research added “Have a shared infrastructure for work products and source code (configuration management)”. These codification practices are not perceived as useful as the practices that focus on personalization. Rather than a focus on codification, the respondents tend to rely on other forms of communication in which practitioners are linked directly to each other; examples include sharing knowledge during meetings or through other communication mechanisms such as email or mailing lists.

Secondly, we can relate the observed increase in perceived usefulness of certain AKM practices with the patterns observed at other practices. E.g., it is important to meet each other at the start of the software development project, during its kick-off. This, in turn, may reduce the importance of knowing who is who across the project; a kick-off meeting already caters for this. In addition, when people already know each other, they are more willing to respond to each other’s e-mail or provide each other with information through a mailing list. Hence, we conjecture that a temporal relationship exists between the AKM practices (“first do a, because it next reduces the need for b”).

Analysis of the questionnaire results shows that no significant differences in the perceived usefulness can be identified depending on the number of sites. However, as noted before, we did observe a difference in perceived usefulness of practices at respondents participating in software development projects that involve three sites.

6.3. What’s Special about Three Sites?

We validated the results described in Sects. 6.1 and 6.2 with key representatives of the IT service provider during the IT service provider’s annual event for Dutch software and systems engineers.

Of the 102 participants to this event, 15 joined a session that was organized for the purpose of the validation. We presented the findings as listed in Sect. 6.1 and 6.2 and conducted a facilitated group discussion on these findings. We list the results below:

- Projects with four or more sites appear to be more
organized. E.g., according to the participants, these projects need to organize communication across sites, because it will be more complex and expensive when it is not. Hence, communication often occurs through a “hub-and-spoke” model, which results in complexity $O(n)$ instead of $O(n^2)$, where $n$ is the number of sites involved.

- Traveling does not scale when the number of sites involved in the software development project increases.
- Software development projects are initially thought to be set up either with more than three sites or with only one or two sites. Projects with three sites typically evolved to that stage in the course of the project, i.e. they started as a one- or two-site project. Following that, the main conclusion of the participants to the validation is that software development projects with three sites organized work in a different way than projects with a different number of sites.

To validate the above statements, we directly contacted the 18 respondents that were involved in a software development project with three sites and the 11 respondents involved in projects with four sites. We asked them a number of questions to investigate further characteristics of these projects:

- How was the software development project organized when it started? What was the size and number of sites at the start?
- How is work currently divided in the software development project? Is there a central site that divides the work across the three sites?

We collected responses from three different software development projects that involved three sites and one project that involved four sites.

We learned that two software projects with three sites did not start as such but evolved to that stage after having run for several months with two sites. The two sites that were involved at the start of the project included the customer’s location and one of the development centers of the IT service provider near the customer’s location. After a couple of months, project management decided to move software development activities to the service provider’s off shore development centers. At that time, communication was not structurally reorganized, which resulted in less communication between the sites. In addition, the practitioners involved in software projects with three sites indicated an increased perceived usefulness for some AKM practices because they encountered several problems in their project. The practitioners indicated that the AKM practices were not implemented but, rather, thought that applying these practices would help in addressing these problems.

Conversely, the software development project with four sites was designed as such at the start of the project and remained stable in number of sites since then. Most AKM practices were either implemented or discarded on beforehand; traveling does not scale with four geographically distributed development sites, so other measures had been taken.

In conclusion, software development projects that encounter an increase in number of sites should be keen to implement additional AKM practices in time to prevent problems from surfacing.

6.4. Limitations

We list possible limitations to our study by analyzing the internal validity and external validity following [21, 29].

**Internal validity** – Numerous other factors than the number of sites of a software development project may affect the perceived usefulness of AKM practices. Although in this research we primarily focus on the number of sites, we validated the results with the organization in which other, underlying reasons were identified.

**External validity** – We conducted our questionnaire at one organization. We asked the respondents selected from this organization to answer the questionnaire with a certain software development project in mind. When this questionnaire is repeated over time, the same group of respondents may be involved in different software development projects and answer the questions according to that project. Given the nature of our population (the complete population of Dutch architects of the service provider) and the variety of software development projects encountered in this questionnaire, we do not expect significant deviations. However, when the questionnaire is completed by another organization, different results may be obtained.

7. Conclusions and Future Work

We have conducted empirical research on the relation between the usefulness of architectural knowledge management practices in global software development. Specifically, we related the usefulness of these practices to the number of sites involved in several GSD projects at a large IT service provider in the Netherlands. We sent out a questionnaire to the Dutch architects of this organization to obtain our results. Using the questionnaire, we related the number of sites to the perceived usefulness of nine practices for AKM in GSD that were identified in earlier research [8]. Seven of these nine practices support a personalization strategy towards knowledge management, the remaining two support a codification strategy towards knowledge management.
The AKM practices in general are perceived as useful (about 40% of the respondents regard the GSD AKM practices as useful). Personalization practices in general are perceived as more useful than codification practices. Some AKM practices show an increase in usefulness as the number of sites increases. This increase peaks at software development projects that involve three sites. Other practices are regarded as useful regardless of the number of sites involved. We learned that the main reason for the increase is that practitioners involved in projects with three sites have a need for additional AKM practices because certain problems are encountered. Several three-site software development projects were not started with three sites but evolved into that stage after having run for some time with two sites. The high perceived usefulness for some AKM practices in three-site software development projects denotes a need to have these practices implemented proactively to prevent problems with architectural knowledge sharing from arising.

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 GRIld For inFormation about architectural knowledge.

We thank Rahul Premraj for his useful advise in the statistical analysis.

References


Appendix: Graphs of the Perceived Usefulness of AKM Practices

This appendix lists the results of our study. For each practice, we indicate the number of respondents that indicated their perceived usefulness (N) and Spearman’s rho (\(\rho\)) as a measure of the correlation between the practice and the number of sites and the perceived usefulness of the practice, including its significance (p-value).
Figure 5. Email, mailing list or telephone to quickly get information. $N = 88$, $\rho = -0.015$, $p$-value $= 0.886$

Figure 6. Have a clear project structure with clear communication responsibilities. $N = 88$, $\rho = -0.125$, $p$-value $= 0.245$

Figure 7. Traveling to other project locations. $N = 88$, $\rho = 0.154$, $p$-value $= 0.151$

Figure 8. Knowing who is who across the project (directory). $N = 88$, $\rho = -0.041$, $p$-value $= 0.702$

Figure 9. Have a single repository for architecture artifacts. $N = 88$, $\rho = -0.088$, $p$-value $= 0.415$

Figure 10. Have a shared infrastructure for work products and source code (configuration management). $N = 88$, $\rho = 0.019$, $p$-value $= 0.860$