Software Architecture Review: The State of Practice

Muhammad Ali Babar, *Lero—the Irish Software Engineering Research Centre*
Ian Gorton, *Pacific Northwest National Laboratory*

Architecture reviews are an effective way of ensuring design quality and addressing architectural concerns. However, the software engineering community rarely adopts the methods and techniques available to support disciplined architecture review processes.

You are about to sign off on the software architecture of a multimillion dollar software-intensive system. What assurance do you have that the architectural decisions underpinning the design are the right ones to deliver a system that meets the required business goals? How sure are you that the project will not be delayed through downstream rework—or even fail—due to inappropriate architectural choices? Do you know whether all system stakeholders have confidence in the proposed solution? Are "best endeavors" a good enough basis for accepting an architectural design?

Architecture reviews are an effective way of ensuring design quality and addressing architectural concerns. The principal objectives of a software architecture review are to assess an architecture's ability to deliver a system capable of fulfilling the quality requirements and to identify potential risks.[1]

To support organizations performing reviews, several working groups, including Software Architecture Review and Assessment, have produced reports[2-4] summarizing architecture review best practices. However, apart from reports from organizations such as AT&T, Avaya Labs, and Lucent Technologies,[3,4] little is known about the state of architecture review practice in the broader software industry. To investigate this issue, we administered a survey to determine the state of practice in the software engineering community.

ARCHITECTURAL REVIEW SURVEY

To our knowledge, ours is the first large-scale research project to determine current industrial practices in architecture review. In our cross-sectional study,[5] we administered a questionnaire to obtain self-reported qualitative and quantitative information from software practitioners in a range of organizations. We designed the survey questionnaire based on

- an extensive literature review and our previous work on comparing architecture review approaches,[6]
- our experiences in designing and evaluating architectures for large-scale systems, and
- in-depth discussions with 20 experienced architects from a range of organizations.

The questionnaire consisted of 20 questions to investigate architecture review practices and six questions on the demographics of the respondents and their organizations. The questionnaire underwent a rigorous review process on the format of the questions, suitability of the scales, understandability of the wording, number of questions, and the length of time required by experienced software architects to complete. We also conducted a pilot study and focus group discussion to further refine the survey.[7] Pilot study data was not included in the main survey.

Because of the way we had to determine survey participants, our sample is not random. Obtaining a random

Published by the IEEE Computer Society

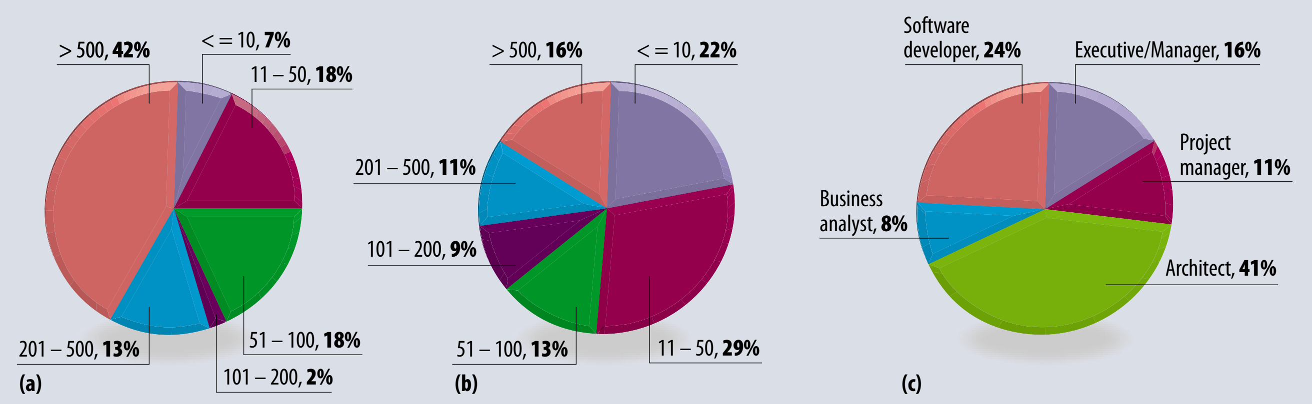

Figure 1. Demographics: (a) organization sizes, (b) organization sizes based on people working in software-engineering-related jobs, and (c) respondents' job positions.

sample is almost unachievable in software engineering surveys because the community lacks thorough demographic information about populations of interest.[8] Thus, we used a nonprobabilistic sampling technique known as convenience sampling.[5] The major drawback of this technique is that the results are not statistically generalizable to the target population.

We selected 235 participants from our professional contacts and industrial referrals. The invitation e-mail included the survey questionnaire and requested that recipients return the completed questionnaire by e-mail. After two reminders one month apart, we received 88 responses. We decided to remove two responses from the analysis because several questions were not answered.

DEMOGRAPHICS

The survey respondents' companies vary in size from small (10 employees or fewer) to large (more than 500 employees). Forty-two percent of respondents stated that their companies employ more than 500 people, as Figure 1a shows. Similarly, the number of people employed in software-engineering-related positions in the respondents' companies ranges from small (up to 10 people) to large (more than 500 people), as Figure 1b shows. Thirty-six percent of the respondents' organizations are certified either for ISO 9001 quality management standards or at different levels of the Capability Maturity Model.

The survey participants also reflect a wide range of positions in their organizations. Figure 1c summarizes the respondents' job position data. A majority are architects (41 percent) or developers (24 percent). The median number of years the respondents were in their current position, involved in software development, and involved in architecture design and architecture review activities are 3, 8, and 3.5, respectively. These demographics demonstrate that the findings of this survey are based on the data gathered from practitioners who are experienced in software architecture design and reviews.

SURVEY RESULTS

Contemporary software architecture research increasingly emphasizes the importance of having a defined review process. Hence, we were interested in knowing whether the respondents' organizations have a defined process to review architectural decisions or if reviews are performed informally based on organizational or personal experience. A related question asked which types of techniques—such as scenarios, checklists, and prototypes—organizations usually use during architecture reviews.

When asked about the nature of the architecture review process, 56 percent of respondents described their organization's review process as informal, 41 percent reported a formal process in place, and 3 percent were not sure, as architecture review processes varied from project to project.

The two most common techniques applied to review an architecture are experience-based reasoning (83 percent) and prototyping (70 percent). Many respondents also mentioned scenarios (54 percent) and checklists (40 percent). Given the popularity of mathematical-model-based review approaches in mainstream software engineering conferences and journals, we wanted to know the extent of industrial use of these methods. Only 5 percent of respondents mentioned that they use mathematical models to evaluate architectural decisions in their organizations. Figure 2 shows the percentages of responses for each review technique.

We also asked about the criteria that companies used to select a certain architecture review technique. Respondents mentioned various technical, managerial, organizational, social, and business-related factors that influence the choice to use a particular technique. The main factors identified are

- relevant quality attributes for the system under review, such as performance, usability, and security;
- size and complexity of the system under review;

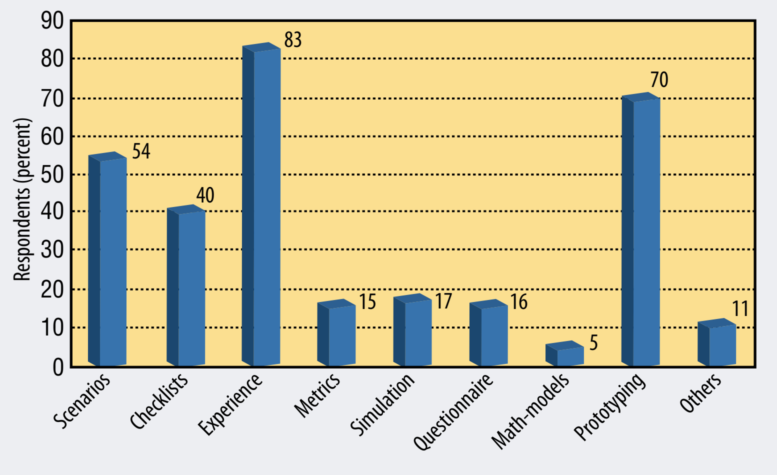

Figure 2. Techniques used for architecture evaluation.

- availability of staff specializing in a certain review technique;
- development time frame and budget;
- customer requirements;
- organizational politics; and
- architect or project manager prerogative.

However, no respondent mentioned having an organizational policy, standard, or guideline for selecting review techniques.

The architecture review research community has developed several dedicated methods to support the review process. Among these methods, Scenario-Based Architecture Analysis Method (SAAM),[1] Architecture Level Maintainability Analysis (ALMA),[9] Architecture Tradeoff Analysis Method (ATAM),[1] and Performance Analysis of Software Architecture (PASA)[10] are discussed in a large volume of software engineering literature. Moreover, researchers have performed several comparative studies of these methods,[6,11,12] which shows that this area is becoming relatively mature and rich.[13]

We wanted to know how many practitioners were aware of and used the widely published architecture review methods. When asked whether they had heard of these methods, 24 percent knew about SAAM and ATAM, 17 percent were aware of PASA, and only 6 percent had heard of ALMA. The response to the question of whether the respondent's organization uses any of these methods was low. Five percent of respondents mentioned SAAM and ATAM, while 1 percent mentioned PASA and ALMA. However, these respondents all indicated that their organizational processes are derived from these methods; they do not use them "out of the box."

Tool support

Architecture review is a human- and knowledge-intensive process, with many tedious and time-consuming tasks including collecting, documenting, and managing the information relevant to architectural decisions and review outcomes. While some organizations have proprietary tools to support architecture reviews,[4] little specialized tool support is available.[6] We were therefore interested in determining the types of tools the survey respondents used during the review process.

Thirty-five percent of the respondents reported they use tools for capturing and presenting architecture description. The most common tools reported were Rational Rose and Microsoft Visio. Nineteen percent of the respondents use an experience repository, and 17 percent use groupware tools. This relatively low level of tool usage indicates an opportunity for specialized tools that capture design artifacts along with design rationale, outcomes, measurement, and administrative information. Such tools could be invaluable for increasing the impact and efficiency of architecture reviews.[1,6]

One of the most frequently cited problems in conducting successful architecture reviews is unavailability or inadequacy of architecture documentation.[14] Given the importance of architecture documentation as a vehicle for communication among stakeholders, we were interested in determining how organizations deal with this issue. As Figure 3a shows, when asked whether their organizations have a clearly defined standard to document and maintain software architecture, 51 percent responded positively.

Next, we asked about the means of documenting software architectures. As Figure 3b illustrates, 83 percent of respondents use various modeling notations, the most common of which was the Unified Modeling Language (66 percent of the 83 percent using modeling notation). Sixty-seven percent reported using architectural views. A majority of respondents (78 percent) reported that their organizations include the main architecture requirements as part of the documentation. Sixty-seven percent also reported documenting assumptions and constraints. Twenty-two percent of respondents reported the use of design patterns, detailed functional specifications, and in-house templates for architecture documentation.

Review practices

The next set of questions in the survey focused on people-related issues in architecture review processes. A strongly recommended practice is using independent experts (external to the project or organization) to review an architecture.[3,4] When asked who is normally responsible for architecture review, the majority reported that architects (63 percent) and the design team (55 percent) perform this task. As Figure 4a shows, only 9 percent mentioned the involvement of an independent team outside the project, and only 3 percent mentioned the involvement of external consultants. This data reveals that for our respondents, architects and their design teams normally review architecture without any significant involvement of external experts—a practice that might reduce the impact of the review.

Another recommended practice for institutionalizing architecture reviews is having a dedicated group (physical or virtual) to support reviews. Some organizations call this the *architecture review board*.[4] When asked about the existence of a dedicated review team, only 29 percent stated that their organizations assign dedicated teams to support architecture review processes. As Figure 4b shows, 68 percent stated that their organizations do not have architecture review teams.

David L. Parnas and David M. Weiss[15] regard the presence of inappropriate people in a design review session as a major problem with the conventional design review

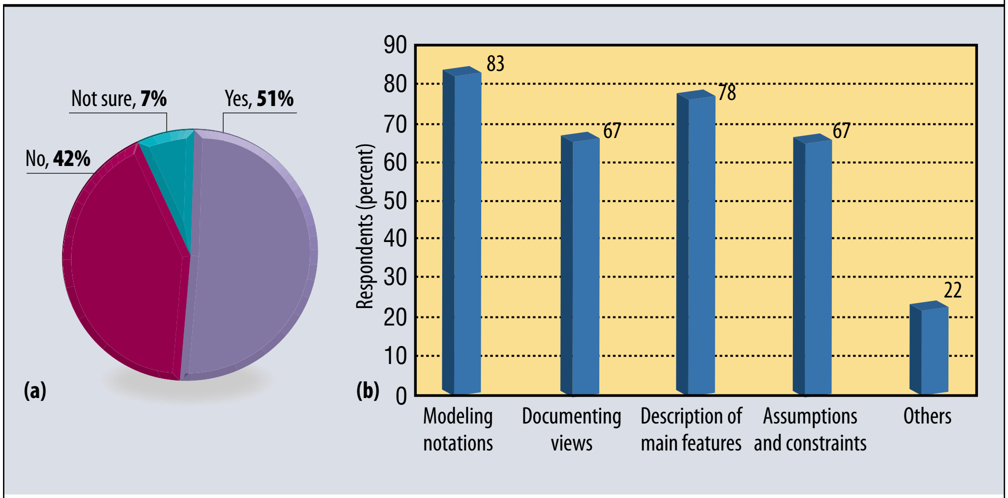

Figure 3. Documenting architecture: (a) organizational standards and (b) means of documentation.

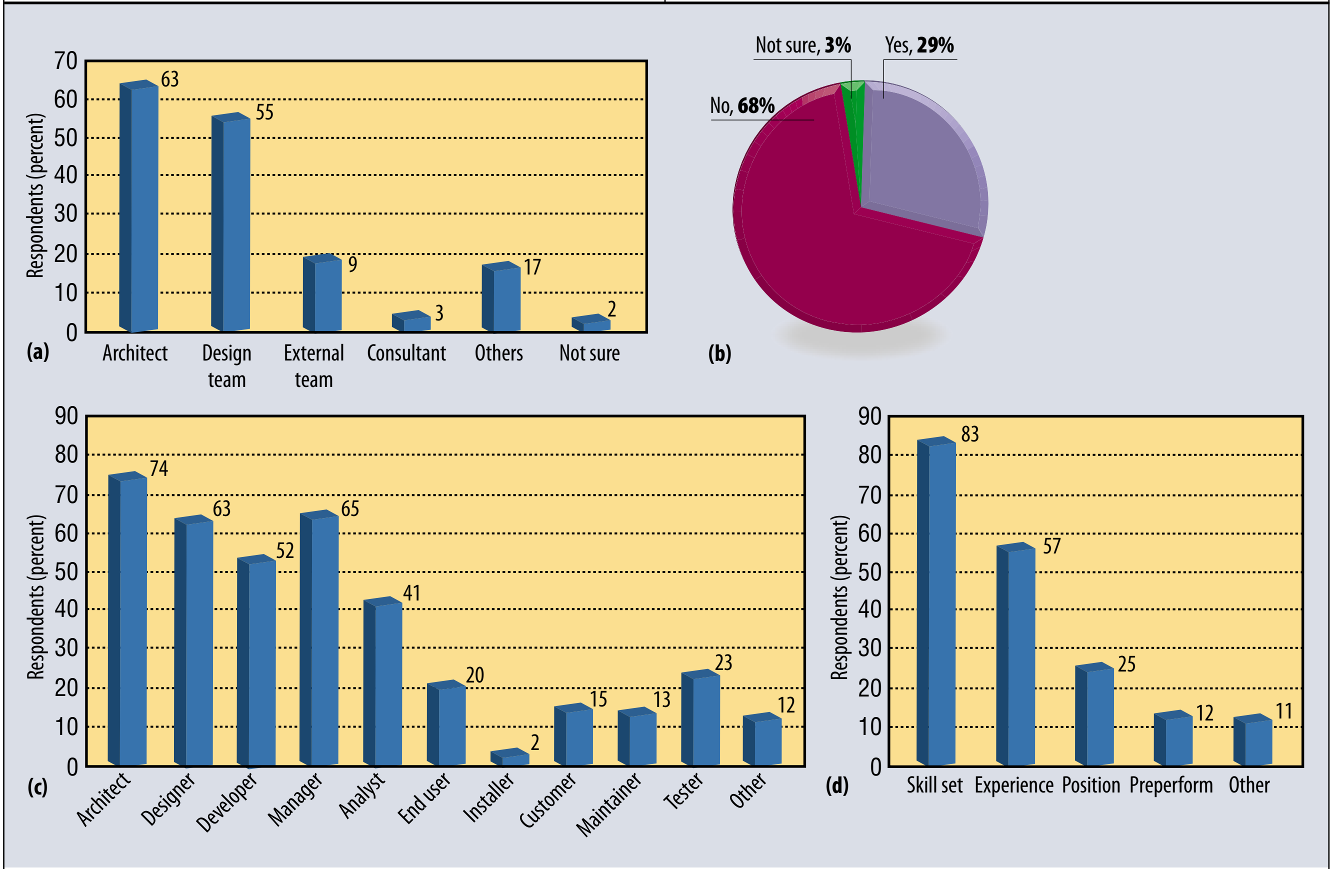

Figure 4. Stakeholders in architecture reviews: (a) reviewers, (b) percentage of organizations with a dedicated team to review architecture, (c) stakeholders participating in review, and (d) criteria used to involve stakeholders.

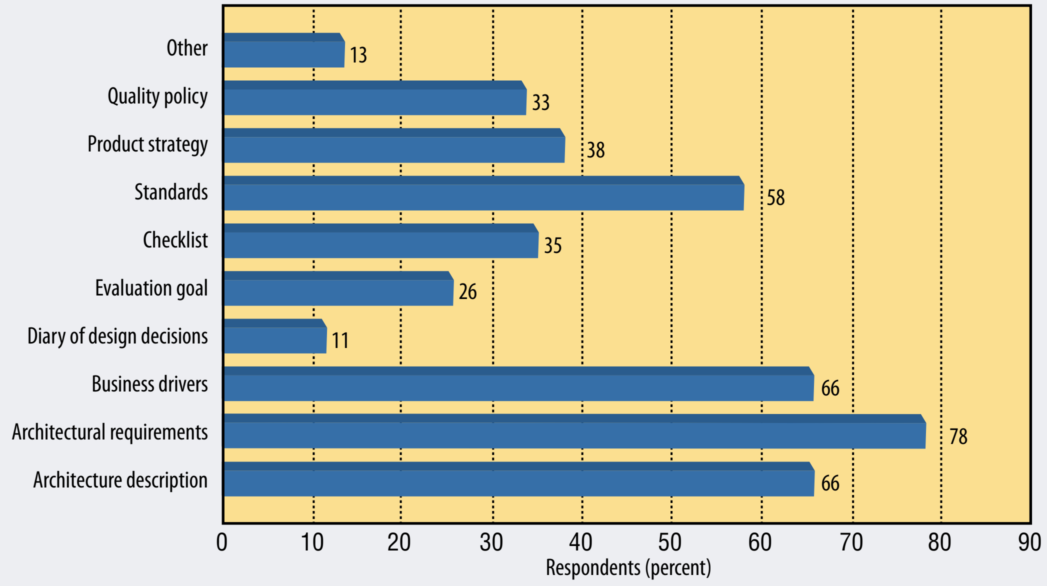

Figure 5. Architecture review inputs.

approach. This is also true with architecture review sessions. Hence, we asked which stakeholders are normally involved in architecture reviews and what criteria are used to select them. We presented a list of classes of stakeholders that are recommended for involvement in architecture reviews and asked the respondents to select all that apply to their organizations' normal practice. As Figure 4c shows, the responses to this question revealed that only certain stakeholder roles—architect (74 percent), manager (65 percent), designer (63 percent), and developer (52 percent)—are commonly involved in architecture reviews.

The data in Figure 4c also highlights that stakeholders such as analysts, maintainers, installers, and testers are not usually involved in architecture reviews. Other important classes of stakeholders often left out of architecture reviews are end users and customers. It would therefore seem that the potential for valuable input from a broader set of stakeholders during a review is being lost in many organizations that perform architecture reviews.

Our next question was about the factors that influence the selection of participants for architecture reviews. As Figure 4d shows, the respondents stated that suitable skills (83 percent), experience in architecture reviews (57 percent), organizational position (25 percent), and previous performance on reviews (12 percent) usually influence the decision to involve someone in a review.

Review inputs

Reviewers need several types of information to understand and review a software architecture. The architecture review research community has identified the types of information required as input for an architecture review process to proceed.[2,4] When asked about the usual inputs for architecture reviews in participants' organizations, the three most commonly reported artifacts were, as Figure 5 shows, architectural requirements (78 percent), architecture description (66 percent), and business drivers (66 percent)—the same three inputs researchers consider the most important for an effective review.[2,14]

Review implementation stage

Organizations conduct architectural evaluations at different stages of the development life cycle. Gregory Abowd and colleagues[3] describe three common project phases for architecture reviews:

- early—after initial and high-priority architectural decisions have been made,
- middle—after elaboration of architecture design but before implementation starts, and
- post-deployment—after the system is in production.

Joseph Maranzano and colleagues[16] also reported two forms of architecture reviews: architectural discovery (early) and architecture review (before coding). We wanted to know the software development stage at which participants' organizations review software architecture. The responses to this question were:

- early stage—80 percent
- middle stage—34 percent
- post-deployment stage—15 percent
- system reengineering stage—28 percent

- system acquisition stage—11 percent

The common use of early-stage architecture reviews aligns with our experiences and discussions with architects. However, an interesting finding is that only a few respondents reported that they review architectures during the system acquisition stage. Using COTS components and packages is a known high risk as they might not be compatible with the system's requirements and environment and can cause integration problems.[17] We are therefore surprised that architecture reviews are not more prevalent during acquisition activities.

Review benefits

The final part of the survey was intended to determine the perceived benefits of architecture reviews. We presented an extensive list of benefits of architecture reviews found in the literature, and participants could select as many statements as they wished. Table 1 provides the list of those statements and the number of responses, listed in descending order.

Table 1. Perceived benefits of architecture review.

Benefits/goals of conducting architecture review	Responses
a. Identifying potential risks in the proposed architecture	76 (88%)
b. Assessing quality attributes (for example, scalability, performance)	66 (77%)
c. Identifying opportunities for reuse of architectural artifacts and components	62 (72%)
d. Promoting good architecture design and evaluation practices	55 (64%)
e. Reducing project cost caused by undetected design problems	54 (63%)
f. Capturing the rationale for important design decisions	51 (59%)
g. Uncovering problems and conflicts in requirements	51 (59%)
h. Conforming to organization's quality assurance process	47 (55%)
i. Assisting stakeholders in negotiating conflicting requirements	37 (43%)
j. Partitioning architectural design responsibilities	34 (40%)
k. Identifying skills required to implement the proposed architecture	34 (40%)
l. Improving architecture documentation quality	34 (40%)
m. Facilitating clear articulation of nonfunctional requirements	27 (31%)
n. Opening new communication channels among stakeholders	27 (31%)

Apart from the perceived benefits of conducting architecture reviews, we were also interested in knowing whether respondents attempt to quantify the benefits of architecture reviews. When asked about this issue, 92 percent of participants stated that their organizations do not quantify review benefits. Of the 8 percent who said they do, none could state any rigorous process for quantification; rather, they rely on numbers of defects detected or risks types identified.

We also explored whether respondents use any formal mechanism for assessing the cost versus benefit of reviewing architectures. Ninety-one percent of participants stated that they do not formally assess the cost-benefits. Of the 9 percent that do, none of them mentioned the specific techniques used to quantify benefits.

SURVEY FINDINGS

Our survey produced some interesting insights about the state of software architecture review practices in our respondents' organizations:

- The majority of architecture reviews are informal, without a systematic approach guiding the review process.
- Using external reviewers is not a common practice. This can reduce the impact of the review, since the project members responsible for the design are less likely to detect flaws in architectural choices made.
- Practitioners seem more inclined to leverage individual review techniques (for example, scenarios) from the research community rather than using a complete method to support architecture review processes.
- Few organizations have a systematic way to quantify the benefits of architecture reviews. A systematic mechanism for showing the return on investment from reviews could help create a solid business case for institutionalizing architecture reviews.
- Most reviews take place early in a project life cycle. Organizations do not regard system acquisition as an important stage for review.
- Most architecture reviews occur on an ad hoc basis. Ad hoc reviews might not provide long-term process improvement to increase an organization's return on investment.

Overall, the survey results reveal one clear message. Despite the research community's major investigations and published results concerning architecture review processes over the past decade, the impact of this research is far from pervasive among our survey respondents.

We posit two possible reasons for this:

- Technology transfer remains a major challenge. There is limited out-of-the-box process and tool support for organizations that want to start reviews.
- Building a convincing business case to demonstrate the return on investment of architecture reviews is difficult.

This survey provides some initial empirical evidence that these two issues might significantly inhibit organizations that wish to gain the full benefit of architecture review.

LIMITATIONS

Like most surveys in software engineering, our study faced reliability and validity threats. Following the guidelines provided in "Principles of Survey Research,"[5] we put certain measures in place to address these issues. For example, researchers and practitioners rigorously evaluated the research instrument, we tested all the questions in a pilot study, and we ensured respondents' anonymity and confidentiality. However, completely eliminating the possibility of bias error is difficult.

The results might also suffer from nonresponse error. If only those with positive views of architecture reviews responded, the results would be biased. However, several invitees sent us apologies because they were too busy to participate. We also contacted other nonrespondents and found they could not participate because of work commitments.

Geographic location of the respondents—mainly Australia—is another limitation. Although we received some responses from China, India, the US, and Vietnam, we cannot be certain that the findings can be generalized globally.

The main objective of our study was to shed some light on current industrial practices in the area of software architecture reviews. The results indicate that while reviews are undertaken in many of our respondents' organizations, there is considerable room for improving review practices to increase their impact. Overall, the results of our survey clearly show that there is much work to be done in transferring the outcomes of software architecture review research into widespread practice. To accelerate technology transfer, existing and new architecture review approaches and techniques should be adequately validated on industrial projects of various scales and in different application domains to refine and extend research results.

The survey results will direct our ongoing research in this area. We are especially interested in how design knowledge repositories and groupware technologies can support architecture reviews. We hope that by providing tailored tool support for architecture reviews, organizations might more easily reap their potential rewards.

Acknowledgments

We thank the participants in this study for their interest and time. Lero is supported by Science Foundation Ireland under grant 03/CE2/I303_1.

References

1. P. Clements, R. Kazman, and M. Klein, *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley, 2002.
2. H. Obbink et al., *Software Architecture Review and Assessment (SARA) Report*, tech. report, SARA W.G., 2001.
3. G. Abowd et al., *Recommended Best Industrial Practice for Software Architecture Evaluation*, tech. report CMU/SEI-96-TR-025, Software Eng. Institute, Carnegie Mellon Univ., 1997.
4. J.F. Maranzano et al., "Architecture Reviews: Practice and Experience," *IEEE Software*, vol. 22, no. 2, 2005, pp. 34-43.
5. B. Kitchenham and S.L. Pfleeger, "Principles of Survey Research," Parts 1 to 6, *SIGSOFT Software Eng. Notes*, ACM, 2001-2002.
6. M. Ali Babar, L. Zhu, and R. Jeffery, "A Framework for Classifying and Comparing Software Architecture Evaluation Methods," *Proc. 15th Australian Software Eng. Conf.*, IEEE CS Press, 2004, pp. 309-319.
7. M. Ali-Babar, L. Bass, and I. Gorton, "Factors Influencing Industrial Practices of Software Architecture Evaluation: An Empirical Investigation," *Proc. 3rd Int'l Conf. Quality of Software Architectures* (QoSA), Springer, 2007, pp. 90-107.
8. T.C. Lethbridge, S.E. Sim, and J. Singer, "Studying Software Engineers: Data Collection Techniques for Software Field Studies," *Empirical Software Eng.*, vol. 10, no. 3, 2005, pp. 311-341.
9. P. Bengtsson et al., "Architecture-Level Modifiability Analysis (ALMA)," *J. Systems and Software*, vol. 69, nos. 1-2, 2004, pp. 129-147.
10. L.G. Williams and C.U. Smith, "PASA: A Method for the Performance Assessment of Software Architecture," *Proc. 3rd Workshop Software Performance*, ACM Press, 2002, pp. 179-189.
11. R. Kazman et al., "A Basis for Analyzing Software Architecture Analysis Methods," *Software Quality J.* vol. 13, no. 4, 2005, pp. 329-355.
12. L. Dobrica and E. Niemela, "A Survey on Software Architecture Analysis Methods," *IEEE Trans. Software Eng.*, vol. 28 no. 7, 2002, pp. 638-653.
13. R. Kazman, L. Bass, and M. Klein, "The Essential Components of Software Architecture Design and Analysis," *J. Systems and Software*, vol. 79, no. 8, 2006, pp. 1207-1216.
14. R. Kazman and L. Bass, "Making Architecture Reviews Work in the Real World," *IEEE Software*, vol. 19, no. 1, 2002, pp. 67-73.
15. D.L. Parnas and D.M. Weiss, "Active Design Reviews: Principles and Practices," *J. Systems and Software*, vol. 7, no. 4, 1987, pp. 259-265.
16. J. Maranzano, *Best Current Practices: Software Architecture Validation*, tech. report, AT&T, 1991.
17. A. Liu and I. Gorton, "Accelerating COTS Middleware Technology Acquisition: The i-MATE Process," *IEEE Software*, vol. 20, no. 2, 2003, pp. 72-79.

Muhammad Ali Babar *is a senior researcher at Lero—the Irish Software Engineering Research Centre. He received a PhD in computer science and engineering from the University of New South Wales, Australia. Ali Babar is a member of the IEEE Computer Society. Contact him at malibaba@lero.ie.*

Ian Gorton *is the chief scientist for Pacific Northwest National Laboratory's Data-Intensive Computing Initiative. He received a PhD in computer science from Sheffield Hallam University. Gorton is a member of the IEEE Computer Society. Contact him at ian.gorton@pnl.gov.*