An Approach That Stimulates Architectural Thinking during Requirements Elicitation: An Empirical Evaluation

Preethu Rose Anish[1], Maya Daneva[2], Smita Ghaisas[1] and Roel J. Wieringa[2]

[1]TCS Research, India
[2]University of Twente, Enchede, The Netherlands

Keywords: Empirical Evaluation, Grounded Theory, Requirements Elicitation, Software Architecture.

Abstract: In many global outsourcing projects, the software requirement specifications (SRS) are often orchestrated by requirements analysts who have sufficient business knowledge but are not equipped to ask the kind of questions that are needed to unearth architecturally relevant information from the customer. Often, the resultant SRS therefore lacks some critical details needed by software architects to make informed architectural decisions. To remedy this, the software architects either make assumptions or conduct additional stakeholder interviews resulting in expensive refactoring efforts and project delays. Using an empirical approach, we have designed an approach of using architectural knowledge that can serve as a communication medium between requirements analyst and software architects. In this paper, we present a detailed empirical evaluation of our proposed approach, with practitioners from real-world organizations. Using two studies, we found that in the experience of the participating practitioners, the approach is relevant, easy to use and effective.

1 INTRODUCTION

Requirements engineering (RE) is the early phase of software development projects in which the system requirements in the form of both functional as well as non-functional requirements are developed and managed. In global distributed software engineering and outsourcing projects, communication between Business Analysts (BAs) and Software Architects (SAs) mostly takes place through a Software Requirements Specification (SRS) and expertise is not shared. The SRS resulting out of these RE activities therefore often lack the details needed by software architects (SAs) to make informed architectural decisions. In the absence of such details, the SAs either make assumptions, which incorrect could lead to expensive refactoring efforts or go back to the business analysts (BAs) for clarifications resulting in project delays. Asking BAs to provide architecturally richer specification may seem like a good idea, but is going to be ineffective, given that BAs lack the technical architectural knowledge needed to ask the kind of questions that are needed to extract architectural details from the customer. This is typical in large-scale software engineering, global development and outsourcing projects where

communication between BAs and SAs mostly takes place through an SRS and expertise is not shared. This problem has been well acknowledged by other researchers as well (Li, 2013, Chen, 2013, Gross, 2012).

As a solution to this problem, we have developed an approach (Anish et al., 2015, 2016) that leverages the knowledge of experienced SAs and make it available to the BAs so that they are equipped to elicit a more complete set of requirements that feeds sufficient information into the architectural design process. In (Anish, 2017), we presented our initial research plan for an empirical evaluation of our approach. In this paper, we present the detailed design, executions and results of our systematic empirical evaluation study. In particular, we intend to investigate three aspects namely the *ease of use*, *effectiveness* and *relevance* of our approach. In our research design, we first evaluate the *ease of use*, *effectiveness and relevance* from the perspective of practicing BAs. By referring to the *ease of use* concept originally published in (Davis, 1989), we measure *ease of use* by gauging how easy it is for the BAs to use the approach as a part of their requirements gathering exercise, whether they find it easy to adapt to this new way or do they consider this
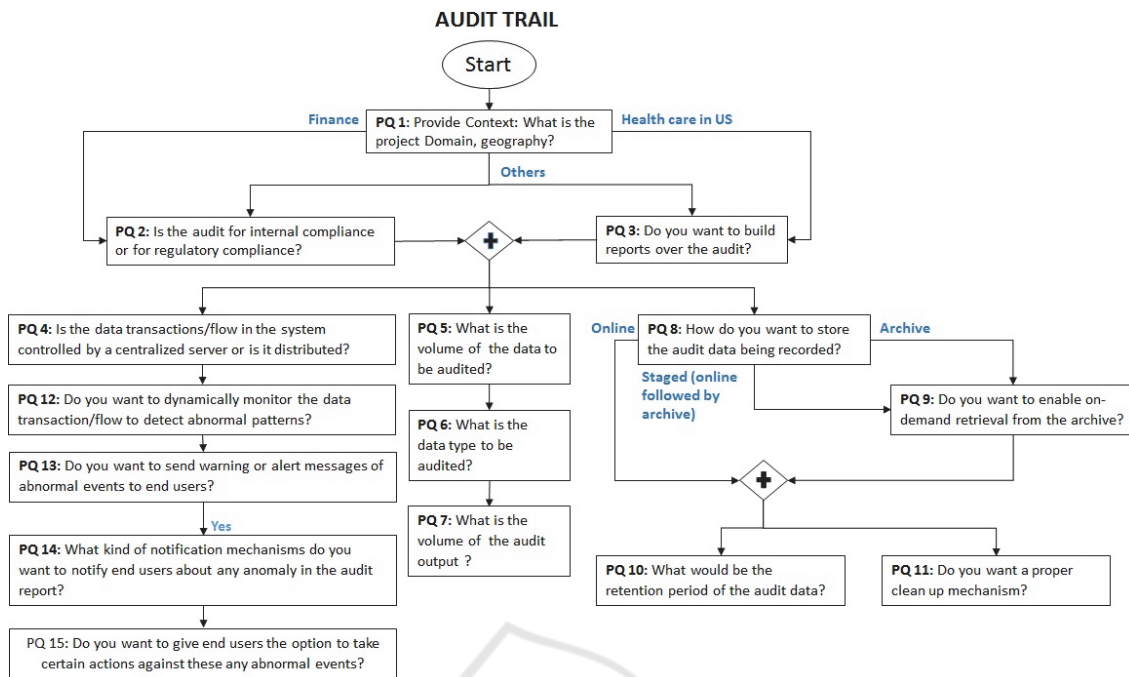
AUDIT TRAIL

Figure 1: PQ-flow for Audit Trail.

as a paradigm shift that they are not able to relate to. By *effectiveness*, we intend to investigate the degree to which our approach is successful in producing a desired result i.e. assist the BAs in unearthing architectural information from the client during requirements gathering. By *relevance,* we mean to investigate whether the BAs find the approach important to their requirement gathering practices and would add value to it. Furthermore, regarding examination of *generalizability*, we include the perspectives of practising SAs. For examining generalizability, the strategies described in (Anish et al., 2015) are considered.

The rest of the paper is organized as follows: In Section 2 we provide definitions of key terms used in our approach, Section 3 provides background and presents details on the approach subjected to evaluation. Section 4 is on our research questions, Section 5 is on research methodology and research process, Section 6 presents discussions while Section 7 concludes the paper.

2 DEFINITIONS

As terminology is not uniform across all authors in the field of RE and software architecture, we define some key terms here. We consulted definitions from different sources such as IREB (https://www.ireb.org/en) and for our purpose, we use the following working definitions. A requirement is called *architecturally significant* if it has a measurable impact on the architecture of the software system. A *functional requirement* (FR) is a desired behavior triggered by some event or condition change, and delivering some desired output of the system. Any other desired property of the system is called a *non-functional requirement* (NFR). We thus distinguish architecturally significant functional requirements (ASFRs) from architecturally significant non-functional requirements (ASNFRs). While both FRs and NFRs can have an impact on architectural design, for our purpose, we focus on ASFRs only. The questions that one may ask to extract architectural information are called *Probing Questions (PQs)* and PQs when logically sequenced in dialogs are called *PQ-flows* (Probing Question flows). A *Business Analyst* (BA) is the central role responsible for understanding (from the client) the business or functional aspects of the requirements for IT projects. Based on this, the BA is responsible for creating a detailed SRS to be used in the subsequent phases of project development. A *Software Architect* (SA) is a role responsible for converting the business requirements captured by the BA into architecture and design.

3 BACKGROUND AND THE APPROACH SUBJECTED TO EVALUATION

As indicated in Section 2, both FRs as well as NFRs have a significant impact on the architecture of the software system. However, to scope our research work, we focus only on FRs. We define ASFRs as those functional requirements that have a significant impact on the architecture of the software system. Through a series of interview with SAs from various business domains and geographies (Anish et al., 2015, 2016), we identified various categories of ASFRs. For each of these categories, through a series of interviews and on-line surveys, we created a set of *PQ-flows* (Probing Question flows). An example of the PQ-flows for ASFR category – *Audit Trail* is depicted in Figure 1. *Audit Trail* facilitates auditing of system execution (Anish et al., 2015). An example *Audit Trail* FR is: "*System must record every modification to customer records for audit purposes.*" We currently have 15 such ASFR categories in the knowledge base. Out of these 15 ASFR categories, we have created PQ-flows for 10 categories. The 10 ASFR categories are: *Audit Trail, Batch Processing,*

Business Process State Alert, Print, Report, Search, Localization/Multilingual, Online Help, Third Party Interaction and Workflow. In Table 1, we reproduce from (Anish et al., 2015), the definitions of each of these 10 ASFR categories along with obfuscated examples.

4 RESEARCH QUESTIONS

The purpose of this empirical study is to evaluate the ease of use, effectiveness and relevance of our approach (the PQ-flows), from the perspective of practitioners in the field.

To this end, we set out to find answers to the following research questions (RQs):

RQ 1: To what extent does a BA perceive it easy to use the approach?

RQ 1.1: How easy or difficult is it for the BAs understand the PQs on their own?

RQ 1.2: What kind of effort / training is needed so that the BA can start using the approach on their own?

Table 1: ASFR Categories, their Definitions and Examples (Anish et al., 2015).

ASFR Category	Definition	Example
Audit Trail	Audit Trail facilitates auditing of system execution.	System must record every modification to customer records for audit purposes.
Batch Processing	This includes requirements facilitating batch processing	The disbursement process should be a daily batch process for regular claim pay-outs
Business Process "State" Alerts	This class of ASFR is concerned with notifications, generated often as a part of executing a business process by the respective system.	Workflow is required to send notification to Underwriters.
Print	This includes requirements facilitating support for printing document.	The commission statement should be printed using the package-supplied format.
Report	This category includes FRs that facilitate Report generation.	System should generate reports on complaint register on monthly basis.
Search	This category includes FRs that provide support for Search functionality.	Claims Assessor should be able to search for claim records to be process.
Localization/Multilingual	Localization/Multilingual involves providing support for multiple language.	During the term, the authority may require the contractor to deliver specific communications in other languages.
Online Help	This category includes requirements pertinent to providing online help facility.	An online help facility should be available for claim intimation process.
Third Party Interaction	This category includes requirements that facilitate interaction with third party components.	Once an application has been applied for, it will be exported to ABC back office where it may be processed both automatically by ABC back office system and manually by an underwriter.
Workflow	This includes requirements that provide support to move work items, facilitate reviews and approvals.	The claim is assigned to the claim-handling clerk with the latest 'last assignment' time stamp. It is then passed on to the claim assessor.

RQ 2: How do the PQ-flows of the 10 categories help improve architectural relevance of requirements in a SRS?

RQ 2.1: To what extent are the questions in each category architecturally *relevant* (no superfluous questions), and

RQ 2.2: To what extent are all architecturally relevant questions for each category asked?

RQ 2.3: To what extent are all ASFR categories architecturally relevant for the system being specified?

We conducted two empirical studies: (1) to answer RQ1 (henceforth referred to as Study 1) and (2) to answer RQ2 (henceforth referred to as Study 2). Study 1 takes the perspective of BAs, while Study 2 takes the perspective of BAs and SAs playing the role of clients (called here "pseudo-clients"). The two studies, though different in terms of participants and execution style, build upon each other. Each study's research process is organized into three main phases: Design, Execution and Analysis (Yin, 2014). In the next section, we present each of the studies in more detail.

5 RESEARCH PROCESS AND RESULTS IN OUR TWO STUDIES

5.1 Study 1

In this section, we detail the research methodology, threats to validity and results of Study 1.

5.1.1 Research Methodology

Design. We compared the research methodologies that are most relevant to evaluation studies in SE (Anish et al., 2015). We chose a qualitative interview-based evaluation research method and followed R. Yin's guidelines for case study design (Yin, 2016). Our choice for case study research method was grounded on the suitability of this method to our research context and research perspective, namely the one of practitioners working as BAs and SAs in real-world projects. Furthermore, we chose interviews to obtain a detail-rich, holistic and contextualized description from the participants about the approach. The interview technique was selected for two reasons: (1) it is suitable for inquiry like ours, and (2) the resulting data offers a robust alternative (Saldana,

2013) to more traditional survey methods. We triangulated the data collected from multiple sources (e.g. participants with varied domain expertise, years of experience, educational background). Below, we present the participants and our interview questionnaire that are part of the design of Study 1:

a) Participants. Our research plan for Study 1 included 10 participants who were BAs in a very large company in the sector of IT consulting and outsourcing services. The practitioners were actively engaged in projects in India and Israel. As confidentiality agreements were a premise for the study, we cannot provide any information on the organizations employing these participants. However, as practicing BAs, their professional profiles had the following common characteristics: (1) all of them work on large business information system projects, (2) they had at least 2 years of professional experience as a BA, (3) most of them are experienced in more than one domain, (4) they were employed in large scale inter-organizational projects in which the BAs and the SAs were not co-located (5) all participants are BAs from the vendor's side. The profiles of our 10 participants is described in Table 2. Therein, the first column shows the participant's ID (P1 to P10), the second column indicates the current application domain in which the participant is involved in systems development activities, the third column indicates the country, the fourth column indicates the years of experience as BA, and the last column indicates the educational degrees of the participant. As shown, the practitioners are active in application domains such as insurance, banking, financial management, and telecom. Eight participants have an MBA degree and two hold a bachelor degree. The experience of the practitioners varied from 2 to 18 years. We deliberately searched for diversity across years of experience and domains, in order to assure that our approach is evaluated from participants of various profiles.

b) Interview Questionnaire. As we wanted to collect BAs' feedback, we designed our interview study by (1) composing an interview questionnaire to help the participant structure her response (2) testing the questionnaire with an experienced researcher and implement changes to improve it; (3) doing a pilot interview to check the applicability of the questionnaire in a real-life context; (4) carrying out in-depth interviews according to the finalized questionnaire presented in the Appendix.

Execution. All the BAs were informed in advance of our research goals and the interview process. The

interviews were on a one-to-one basis. The first author of this paper conducted all the interviews. The 10 ASFR categories were shared with participating BAs and they were asked to choose one category that they are most familiar with and one that they are least familiar with. For the two chosen categories, we shared the corresponding PQ-flows and the interview questionnaire. The duration of each interview was between 30 and 60 minutes. The questionnaire included sections designed to collect information about the BA's (i) experience and application domain (ii) understanding of ASFRs and PQ-flows.

Table 2: Details of participants for Study 1.

Participant ID	Application Domain	Country	Total Years of Experience as a BA	Educational Background
P1	Insurance	India	4	B.COM, MBA
P2	Banking	India	3	BE, MBA
P3	Telecom	Israel	13	BSc.
P4	Insurance	India	6	BE, MBA
P5	Insurance	India	6	BSc., MBA
P6	BFSI	India	2	B.COM, MBA
P7	Telecom	India	18	BSc., MBA
P8	Telecom	Israel	10	BSc.
P9	Telecom	India	7	BCA, MCA
P10	Telecom	India	3	BE, MBA

B.COM – Bachelors of Commerce, MBA – Masters of Business Administration, BE – Bachelors of Engineering, BSc. – Bachelors of Science, BCA – Bachelors of Computer Application, MCA – Masters of Computer Application

Analysis. We used qualitative coding of our data (Saldana, 2013), which helps us classify the various reasons as to why BAs perceive a particular category and/or PQs as more difficult or easier than others, what techniques can be adopted to improve the ease of using the approach and make it more effective in assisting the BAs in eliciting architecturally richer specifications.

5.1.2 Threats to Validity

Regarding Study 1, we devised measures to counter the following validity threats (Yin, 2014):
(1) *Researcher's bias:* as the authors were involved in creation of the PQ-flows, there is an elevated risk of passing bias into data collection and analysis. To reduce this risk, the authors let the BA select the category to discuss and freely explain the kind of difficulties felt. The authors took conscious steps to avoid providing any unnecessary information or explanation, except those that the BA asked explicitly.
(2) *Interviewee background:* BAs could vary in terms of collaboration relationships they established with their respective SAs in a project. Some BAs might be more exposed to SAs' work than others. We think however that this threat is minimal because our participants worked in organizations that have standard project delivery process; where knowledge sharing standards and tools are instrumental in keeping SDLC processes consistent across projects in the same domain.

5.1.3 Results

This section presents our findings regarding our RQ 1.

RQ 1: To what extent does a BA perceive it easy to use the approach?

As indicated in Table 1, we received responses from 10 BAs. Our results on sub-RQs in RQ 1 are as follows:

RQ 1.1: How easy or difficult is it for the BAs to understand the PQs on their own?

In our study, we found that the senior BAs (those with 10 or more years of experience) could understand all the PQs on their own. The BAs at the mid-level of experience (5 to 9 years) needed guidance to understand some questions (on average 3 questions) and junior BAs (less than 5 years' experience) needed relatively more guidance (on average 5 questions).

Moreover, we found that the domain expertise did not really have any influence on the understanding of the PQs, which indicated that our PQs are generic across business information system domains. Another factor that affected the result was the BA's educational background. This observation was especially relevant for junior BAs (less than 5 years' experience). Junior BAs with a computer science (CS) background found it easier to understand the PQs than the junior BAs with non-CS background. The BAs attributed this difference to the fact that the BAs with CS background had already taken software architecture course as a part of their educational curriculum and therefore they are familiar with the PQs vocabulary. This indicates that our approach can be used to improve the level of architectural understanding of junior BAs with non-CS background.

RQ 1.2: What kind of effort / training is needed so that the BA can start using the approach on their own?

From the responses received from the BAs, we observed that providing some form of guidance to further elaborate the PQs would improve the understandability of the PQs.

In the words of participant P9, "*elaboration of PQs is important. When we ask questions to the client, they sometime ask us to elaborate it or sometimes they ask counter questions to us. So, we should be well versed with what the PQ is all about.*"

Another participant P7 mentioned, "*We cannot use PQs as a black box. We need to know what it entails*".

Based on the analysis of the interview responses, we found that a guidance mechanism for elaborating PQs could take multiple forms: (1) a one hour self-training module for junior BAs, (2) an embedded self-training module in the tool, (3) a user manual with 'how to start working' steps detailed, (4) some kind of look up facility in the tool as and when required.

5.2 Study 2

In this section, we detail the research methodology, threats to validity and results of Study 2.

5.2.1 Research Methodology

Design. As in Study 1, the design of Study 2 rests on the methodological sources already presented in Section 5.1.1. Study 2 was planned as an interview-based research process with participants and interview questions described in the sub-sections (a) and (b) below. We planned is to ask volunteer BAs, pseudo-clients and SAs (5 each) to simulate a process in which requirements are specified by the BA in consultation with the pseudo-client and is used by the SA to design software. The goal is to observe and analyse simulations in which BAs and SAs use our approach, and to use these observations to answer RQ 2. The design includes a volunteer BA and a pseudo-client who simulates a RE process using our approach, and a volunteer SA who assesses how much the resultant SRS contributes in explicating the hidden architectural requirements. Based on a post-simulation interview with the SAs, we collect their reflections on their experience.

a) Participants. Our research plan for Study 2 included BAs, SAs and pseudo-customers who worked in very large company in the sector of IT consulting and outsourcing services. The BAs and SAs were experienced team members responsible for delivering large projects in varied domains. Any person with more than 20 years of experience working closely with clients to understand their requirements and to architect systems qualified as a pseudo-client. Based on the domain expertise, we grouped the BAs, SAs and pseudo-client. We created five such tuples (T1 to T5). The details of the participants in each tuple is presented in Table 3. As indicated in Table 3, T1 and T2 worked in Insurance domain, T3 and T4 in banking domain while T5 worked in healthcare domain.

b) Interview Questionnaire. Our post-simulation interview questionnaire is developed using the same steps as in Study 1 and is presented in the Appendix.

Execution. We sent the study participation request through email to BAs, pseudo-clients and SAs explaining them our study goal and process. The study execution included the following steps:

(1) We provided the list of 10 ASFR categories along with their description to each of the participating BA.

(2) We asked each BA to choose an ASFR category of her choice based on her expertise and familiarity with the category. If a category was chosen by a BA, we marked the same in the provided list. We told the BAs that for us to ensure validation of more number of categories; we would prefer them to choose a category that is not previously chosen by other BAs. However, we clearly mentioned that this is just our preference and they are free to choose otherwise. To our delight, each BA chose a different ASFR category. The category chosen by each of the five BAs is indicated in Table 3.

(3) The PQ-flow corresponding to the chosen ASFR category was provided to the BA along with detailed instructions on how to use it.

(4) At the meeting between the BA and the pseudo-client, we provided one sample requirement from real life SRS document corresponding to the domain and the chosen ASFR category. The selected requirement corresponding to each of the chosen ASFR category is also presented in Table 3. The BA used the PQ-flow to assist the pseudo-client in elaborating the selected requirement with architectural details and create a much detailed requirement.

Table 3: Details of participants for Study 2.

Tuple Id	BA Exp. (years)	SA Exp. (Years)	Pseudo-client Exp. (Years)	Domain	Chosen ASFR Category	Requirement from SRS
T1	7	10	21	Insurance	Audit Trail	All changes shall be logged so you can see who added/edited/deleted a client and when.
T2	6	9	23	Insurance	Business Process 'State' Alert	A notification shall appear when the user logs onto the system (to inform that the client has converted the quotation to a policy).
T3	6	12	27	Banking	Report	The system shall create a detailed client report at the end of each quarter.
T4	5	8	20	Banking	Batch Processing	On a daily basis, there shall be a batch process that activates the future to-dos, changes the urgency level, and close to-dos.
T5	8	8	21	Healthcare	Search	The system must search in the systems mentioned in the <<document name>> and identify the patient record.

(5) This detailed requirement specific to the chosen ASFR category along with the complete SRS from which this requirement was taken was then given to the participating SA. The SA assessed the elaborated requirement and gave feedback on whether the PQ-flows helped in detailing the requirement further by providing architectural details pertinent to the requirement. In addition to the assessment of PQ-flows, we also provided each SA with the list of all the 15 ASFR categories so that they can comment on the relevance of each of the category.

Analysis. As we collected participants' reflections in the form of qualitative data, we used the coding method similar to Study 1. We expected it to yield codes that explain why the approach worked according to the participant or why they would (or would not) use the approach in their next project and what improvements are needed in the approach to make it practically more relevant.

5.2.2 Threats to Validity

Regarding Study 2, our most important concern is that the simulation is dependent on BA, SA and pseudo-client tuples and as we are relying on volunteers in each role, the relationship between the three is not known to us. However, we relied on professional code of conduct and even if the BA, SA and the pseudo-client have prior working history, we asked them to avoid referring to it during the simulation. Following (Basili & Zelkowitz, 2007), while a simulation in practical settings may be hard to generalize to other context, its key value is in experiencing what in a method works and why it

works (or why not). We take this simulation as a pilot and expect the learning from it to be instrumental in improving our approach and its application scenario.

5.2.3 Results

In this section, we present our findings regarding our RQ 2.

RQ 2: Can the PQ-flows help improve architectural relevance of requirements in a SRS? Below are our findings based on the responses we received from each of the five SAs who participated in the simulation study.

Our analysis of the responses revealed that even though all the SAs found the approach to be very relevant to unearth implicit architectural concerns, the opinion of the SAs on the relevance of an ASFR category or a PQ was highly influenced by application domain and the current project the respective SA was working on. Each SA viewed the ASFR category or the specific PQ from the angle of how relevant it is in her/his current project context. None of the SAs said that the existing ASFRs and/or the PQ-flows are irrelevant. However, they did comment on the relative significance of few of the PQs and ASFRs based on their current project context, changing business trends and the advancements in technology. For example, the SA in T1 with experience in Insurance domain mentioned the following about an ASFR category:

"*If you see the category User behaviour Analysis. This is too specific to applications with consumer-oriented front end. Generally, an Insurance application for Indian clientele would not worry much about this category as even today in India as*

you know most of the insurance policies are bought off line from agents. So we would not worry much about user experience in this case."

From this response we also gauged that in addition to domain, the geography in which project needs to be deployed also has an impact on the relevance of an ASFR Category and/or PQ. For example, a law prevalent in certain geography may advocate certain architectural considerations.

In addition, all of the SAs mentioned that the PQs and ASFR categories would get updated based on the latest trends in the business sector and technology sector. For example, SA in T5 with experience in healthcare domain mentioned the following:

"*With the advent of, say for example, eHealth, mHealth, Telehealth, we would see new ASFR categories and therefore new PQs. Some of the existing PQs may become irrelevant as well*".

Further, SA in T3 talked about block chain technology and how is it reshaping the banking sector and therefore how new ASFR categories and PQs would be needed to address the new architectural needs.

From the responses received from the SAs, it is clear that our approach comprising of ASFR categories and PQ-flows to assist BAs in unearthing implicit architectural concerns during requirements elicitation is relevant. However, the relevance of the existing ASFR categories and the corresponding PQ-flows is heavily dependent on factors such as the project domain, geography and technology and this is subject to evolution based on the specific project context. We consider this feedback important as it sheds light on the possible directions that our work can take to make the repository of ASFR Category and PQs-flows more useful to its users.

6 DISCUSSION

Our approach stimulates architectural thinking during requirements elicitation by equipping the BAs to ask architecturally relevant questions to the customer. We found 15 categories of ASFRs and created PQ-flows for 10 of these categories. In this paper, we presented the empirical evaluation of our approach. Our evaluation has some implications for practitioners and researchers.

For RE practitioners, knowing which categories of ASFR have architectural impact can help the BAs to elicit a more complete set of requirements to feeds

sufficient information into the architectural design process. This would, in turn, help the architects to make informed decisions and can potentially reduce wasted effort caused by the need to rework the solution later in the project. More empirical research, however, is needed to explore the best improvement scenarios that could be implemented in organizations. We therefore consider this as a line for future research.

Our empirical evaluation has some research implications as well. First, it is the starting point for a series of empirical studies in the industrial context in which the PQ-flows were developed in the first place. More empirical data would lead to more generalizable results, which would allow us to draw some recommendations to practitioners on the use of our approach in contexts. Second, we consider the PQ-flows as "living artefacts" that could grow over time; for example, SA in new application domain (e.g. Internet of Things) might come up with additional PQs. Our evaluation research design (grounded on the chosen research methodology) could be used as a blueprint for the empirical evaluation of these new PQ-flows. Third, researchers in RE and SA might find it interesting to compare and contrast our approach with other existing collaborative approaches that help requirements specialists and SAs work together. For example, the approach of Gross et al. (2011), Keim, and Koziolek (2019) could be good candidates for inclusion in such a comparative evaluation.

7 CONCLUSIONS

In this paper, we presented the findings of the empirical evaluation of our approach designed to stimulate architectural thinking during requirements elicitation. This evaluation was intended to gauge the *ease of use*, e*ffectiveness, relevance* and *generalizability* of our approach. From the responses received from the BAs, we conclude that the tool based on the approach should have an embedded self-training module for the BAs. Further, each PQ should include some explanatory text along with it as a ready reckoner for the BA. The BA can look up this text while gathering requirements.

Next, regarding relevance, we observed that the SAs found our approach to be relevant. However, the relevance of the existing ASFR categories and PQ-flows is heavily dependent on factors such as the project domain, geography and technology. For example, the ASFR category *User Behaviour Analysis* is not critical for an Insurance application for

Indian clientele as in India most of the insurance policies are bought off line from agents. This finding also provides answers to the question on generalizability of our approach. The approach itself is generalizable, but the ASFRs and PQ-flows are not. The set of ASFRs and PQ-flows would evolve depending on the project context and therefore a KB with knowledge about the ASFR categories and PQ-flows would support such an evolution.

REFERENCES

Anish, P.R., Daneva, M., Cleland-Huang, J., Wieringa, R.J., Ghaisas, S., What You Ask Is What You Get: Understanding Architecturally Significant Functional Requirements, RE 2015, IEEE Press, 86-95

Anish, P.R., Balasubramaniam, B., Sainani, A., Cleland-Huang, J., Daneva, M., Wieringa, R.J., Ghaisas, S., Probing for Requirements Knowledge to Stimulate Architectural Thinking, ICSE 2016

Anish, P.R., Empirical Evaluation of an Approach that Stimulates Architectural Thinking during Requirements Gathering. In *REFSQ Workshops*, 2017

Davis, F.D., Perceived Usefulness, Perceived Ease Of Use, And User Acceptance of Information Technology, MIS Quarterly; Sep 1989; 13, 3; ABI/INFORM Global pg. 319

Wieringa, R., Daneva, M., Six strategies for generalizing software engineering theories. Sci. Comput. Program. 101: 136-152 (2015)

Yin, R.K., Case study research, Sage, 2014.

Saldaña J. (2013), The coding manual of qualitative researchers (2. ed.). Los Angeles, London, New Delhi.

Basili, V. R., Zelkowitz, M.V., Empirical studies to build a science of computer science. Commun. ACM 50(11): 33-37 (2007)

Gross, A.,Dörr, J., What do software architects expect from requirements specifications? Results of initial explorative studies. TwinPeaks@RE 2012: 41-45

Keim, J., Koziolek, A., Towards consistency checking between software architecture and informal documentation. In *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, March 2019, pages 250–253.

Li, Z., Liang, P., Avgeriou, P., Application of knowledge-based approaches in software architecture: a systematic mapping study, Information & Software Technology, 55, 2013, pp. 777-794

Chen, L., Ali Babar, M., Nuseibeh, B., Characterizing architecturally significant requirements, in IEEE Software, 30(2)2013: 38–45

International Requirement Engineering Board (IREB) Website: https://www.ireb.org/en Last accessed on 25-03-2020

APPENDIX

'About you' question common to both the studies
- Total years of IT experience
- Number of years' experience in working as a software architect /Business Analyst
- Business sector
- Educational Background

Study 1

Q 1. Are all the PQs self-explanatory or do you need explanation?

Q1.1. Please list down the PQs that are not self-explanatory and need further explanation.
Q 1.2. Please suggest how the understandability of these PQs can be improved.

Q 2. What kind of a training module do you think would be required to enable you to use the approach on your own?

Q.2.1 How much of training effort do you think would be required to use this approach on your own?
Q 2.2 How much of a training effort would you be able to put in?

Study 2

Q 1. Are there any superfluous PQs in the given ASFR category? If yes, specify which ones.
Q 2. Are all architecturally relevant questions for the given ASFR category present as a PQ? If no, specify which ones are missing.
Q 3. Are all ASFRs architecturally relevant for the system being specified? If no, specify the missing ones.