

Assessing Effectiveness of Recommendations to Requirements-Related Problems through Interviews with Experts

Aleksander Jarzębowicz, Wojciech Ślesięński

Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Narutowicza 11/12, 80-233, Gdańsk, Poland
Email: olek@eti.pg.edu.pl, slesinski.wojciech@gmail.com

Abstract—Requirements Engineering and Business Analysis are known as very important to software project outcome but also difficult activities, coping with many problems and challenges. The work reported in this paper was preceded by a survey which revealed most common requirements-related problems in Polish IT industry. We addressed ten most frequently reported problems by reviewing the literature for recommendations how to cope with those problems. The resulting set of recommendations is included in the paper. Next, we conducted interviews with three experienced IT analysts asking them to assess effectiveness of particular recommendations, based on their experience. The results show significant differences in assessments and indicate that effectiveness is dependent on contextual factors to a large extent. Our conclusion is that a follow-up work is required to document more recommendations and to annotate them with guidelines about applicability, intended context of use and possible pitfalls.

I. INTRODUCTION

REQUIREMENTS are an essential element of virtually any software project. Regardless of which term is used to describe requirements-related activities – Requirements Engineering (RE) or a more recent, broader term of Business Analysis (BA) - delivery of an effective IT solution depends on identifying and managing business goals and stakeholders' needs. Negligence and flaws in RE/BA processes too often result in unpleasant consequences affecting the whole software project, including its failure and cancellation [1]–[3].

RE/BA is known as a difficult software project area [4]–[6] and as such is a subject of surveys aimed at identification of the most frequent and/or severe problems encountered by practitioners dealing with requirements e.g. [7]–[9]. Also, knowledge about such problems is disseminated by reporting experience [10]. To a large extent, RE/BA problems are therefore known and described in literature since a long time [11]. Yet, they are still present in industrial practice and recognized as causes of project failures even by very recent studies [3]. It raises a question what counter-measures exist to address such problems and how effective they are.

We intended to answer these questions in the context of IT industry in Poland. The first step was to identify relevant

RE/BA problems to be addressed. Despite the fact that a number of surveys were conducted in several countries [7]–[9], it is hard to assume the results would be the same for e.g. different countries or different software development approaches – in fact differences are reported [9] and for that reason surveys on RE/BA problems are replicated in different settings [3].

In 2017 we conducted a survey among IT analysts from Poland to identify RE/BA problems which were most frequently encountered in their professional experience [12]. It provided us with a starting point to the research described in this paper, guided by the following research questions:

- RQ1: What recommendations are known to remedy most frequent requirements-related problems of Polish IT industry?
- RQ2: What is the effectiveness of applying such recommendations in practice?

To answer these questions, first we searched for recommendations by reviewing books on RE/BA and other sources reporting on industrial practice. For each of the top problems from the survey, we found several recommendations about how to deal with them and we documented them. Next, we interviewed 3 experienced analysts asking them to select recommendations they had applied in response to particular problems and to assess whether such recommendations turned out to be effective or not according to their experience.

The main contribution of this paper is the set of recommendations concerning 10 requirements-related problems, provided with assessments of their effectiveness from interviewed experts.

The remainder of the paper is structured as follows. In Section II we review the related work. Section III briefly describes the survey and its results, which provide input to the research study reported here. The study is presented in Sections IV–VI which cover: the search for recommendations and its results (IV), assessment of recommendations effectiveness through interviews with experts (V) and discussion of results and their validity (VI). The paper is concluded in Section VII.

II. RELATED WORK

In our study we address requirements-related problems by interviewing RE/BA experts. Interviews are dedicated to the assessment of the effectiveness of recommendations on how to mitigate these problems. Below we outline other works situated in the RE/BA domain, concerning recommendations to problems or good practices to be followed, identified on the basis of industrial experiences.

Several studies on addressing known RE/BA problems were reported. El Emam and Madhavji [13] interviewed practitioners, which resulted in identifying seven key issues of greatest concern in RE practice at the time and issuing recommendations for improving RE processes w.r.t. these issues. Bjarnason et al. [14] reported a case study of large software development company transitioning towards agile processes. Its employees were asked whether particular agile RE practices mitigate known problems of traditional RE and what new challenges those practices can introduce. De Oliveira Neto et al. [15] looked into the practices influencing both RE and system testing. They used knowledge gathered from interviews with practitioners to map the practices to the challenges encountered in large-scale agile development. Alsaqaf et al. [16] investigated to what extent and how are the non-functional requirements (NFR) included in agile

large-scale distributed development projects. They interviewed practitioners to learn about existing practices as well as associated challenges with the ultimate goal to develop a set of good practices concerning NFRs in that kind of projects.

Recommendations and good practices can also be identified from experience of practitioners, without mapping them to the problems. Hickey and Davis [17] interviewed 9 well-known RE experts to determine which requirements elicitation techniques they would select in various situational characteristics of software projects. As a result, recommendations about the applicability of particular elicitation techniques were derived. Cao and Ramesh [18] focused on agile RE practices and used interviews, observations and documentation reviews to learn about implementations of 7 agile RE practices and associated benefits and challenges. Paul and Tan [19] conducted interviews with expert analysts to gather their opinions about the role of business analyst, his/her contribution to software project success and essential skills.

Our aim was to address particular problems identified in our survey conducted in Poland [12]. Problems, challenges and practices described in the abovementioned sources differed from our case, so no results could be used directly and a dedicated study had to be conducted.

TABLE I.
TOP REQUIREMENTS-RELATED PROBLEMS ACCORDING TO SURVEY RESULTS

ID	Problem	Description	Mean value
P1	Unrealistic expectations of stakeholders	A stakeholder has unrealistic expectations (e.g. large functionality to be delivered in a very short time or at low cost)	2,95
P2	“Obvious” requirements not communicated	Some requirements are so obvious to stakeholders that they do not even mention them	2,80
P3	Scope creep	Constant changes to project scope, modified or expanded requirements (scope creep)	2,80
P4	Too short time for analysis	Too short time for analysis is planned in project schedule	2,75
P5	Stakeholders’ low availability	A stakeholder has little time available to commit to the project	2,73
P6	Solutions issued instead of requirements	Stakeholders describe solutions (e.g. detailed UI design) instead of requirements	2,62
P7	Requirements in the form of change requests only	Stakeholders are not able to express their requirements, they express them later as change requests to working software	2,56
P8	Conflicting requirements	Different stakeholders have mutually contradicting (conflicting) requirements	2,53
P9	Stakeholders ignore business goals	Stakeholders focus on requirements only, they do not define business goals or do not consider them important	2,49
P10	Requirements exceeding project scope	Stakeholders define requirements which are clearly outside project’s scope established earlier	2,44

III. SURVEY AND ITS RESULTS

This section briefly describes the survey and the top problems reported by respondents, which provide input to further research presented in the next sections.

The survey took place in Spring of 2017. We prepared a web-based questionnaire (in Polish), published it and invited

respondents through websites and social network groups for IT analysts. We received 55 answers. As only analysts were targeted, we included a question about respondent’s experience in RE/BA. The distribution of answers was as follows:

- 3 (6%) – less than 1 year;
- 9 (16%) – over 1 but less than 2 years;
- 21 (38%) – over 2 but less than 5 years;

- 14 (25%) – over 5 but less than 10 years;
- 8 (15%) – over 10 years.

The questionnaire included 64 RE/BA problems gathered from the review of literature and the direct communication with a number of analysts [12]. For each problem, the survey respondent was supposed to assess how frequent such problem had been encountered in his/her work experience. The following answers were available: never (0); rarely (1); sometimes (2); often (3); always (4). The numbers in parentheses indicate numerical values that were used when processing and analyzing the survey results. We used them to calculate metrics, which represent frequency of occurrence of each problem and provide a basis for a ranking. Despite the fact that ordinal scale was used to represent answers, we calculate means values, not medians, as for a 5-point scale it is unlikely to spot differences comparing medians.

Table I gives a summary of survey results – 10 problems assessed as most frequent by survey respondents. For each problem, the table includes its artificial ID (to be used as reference in the remainder of the paper), short name of a problem, a longer description (used in the survey questionnaire) and the calculated mean value of answers, which is used as a metric representing frequency.

IV. RECOMMENDATIONS

The search for recommendations addressing the problems listed in Table I included books dedicated to RE: [20]–[23]. We consider them representative, as both internationally recognized items and positions limited to Polish readers are included. Most of them were published in recent years, except the book by Leffingwell and Widrig, which we included due to its influence (number of citations). Additionally, the training materials issued by RE/BA professional associations as well as the on-line courses were reviewed w.r.t. the potential solutions to the abovementioned problems.

Below, we present recommendations we were able to find, divided into sub-sections dedicated to particular problems. Descriptions of the recommendations were shortened and unified when multiple sources mentioned a similar way of dealing with a given problem. Recommendations are assigned the identifiers used in the remainder of the paper. The identifier Rx.y denotes a recommendation number y to a problem number x.

A. P1 - *Unrealistic expectations of stakeholders*

Two major sources of this problem can be identified and, thus, two groups of recommendations are distinguished.

Unrealistic expectations can stem from stakeholders' attitude to issue all requirements they can think of, even those unfounded by business needs or concerning very distant (and uncertain) future. In such case, expectations can be toned down by reducing the number and scope of requirements by the following actions:

- R1.1: Identify business goals as a reference point for requirements.
- R1.2: Identify what does the customer actually expects after the product is deployed (it may not be explicitly articulated without guiding questions).
- R1.3: Verify quality of issued requirements (rationale, unambiguity, feasibility) and do not proceed with requirements of low quality.
- R1.4: Conduct requirements analysis and verify how particular requirements support business goals.

Another possibility is that requirements are consistent with business needs, but their scope is not adjusted to the project constraints. The point of view of at least some of the customers can be to get as much as possible, as fast as possible and at minimal cost, which is a clear violation of the "iron triangle" [24] constraints and is not feasible in real-life projects. To mitigate such situation it is recommended to:

- R1.5: Precisely define product's scope agreed between the supplier and the customer.
- R1.6: Develop a realistic project schedule and manage it during the whole development process.
- R1.7: Estimate development costs (including all relevant categories) and monitor expenses to keep the project within its budget.
- R1.8: Continuously monitor requirements' statuses to be aware which ones are satisfied by the current version of the system under development.

B. P2 - *"Obvious" requirements not communicated*

Stakeholders can assume that some requirements are so obvious that they do not even have to mention them. Unfortunately, it leads to incomplete requirements. Especially non-functional requirements can be omitted this way. Missing requirements are harder to detect than e.g. conflicting ones and thus they can be identified late e.g. during acceptance tests. To prevent this:

- R2.1: Research the literature on the problem domain as well as the similar existing software systems (even before eliciting requirements from human stakeholders).
- R2.2: Make sure that all stakeholders and points of view relevant to the developed system are identified (using e.g. stakeholder map or onion model techniques).
- R2.3: Involve technical experts (on e.g. security, usability) into requirements elicitation and analysis processes.
- R2.4: Use appropriate requirements elicitation techniques, preferably a combination of several techniques including group work (e.g. workshops, focus groups).
- R2.5: When specifying requirements, use Software/System Requirements Specification templates, which include the sections covering various categories of requirements, goals and constraints.

- R2.6: Apply requirements analysis techniques (e.g. checklists, CRUD tables) aimed at the identification of inconsistent, incomplete or missing requirements.

C. P3 - Scope creep

Requirement changes are to be expected in virtually any project, but continuous and uncontrolled changes will probably lead to scope creep. Such situation is considered harmful, especially when such changes are avoidable (e.g. result from communication flaws). Moreover, applying a change usually requires additional resources, but it is often not recognized by the customer. The following recommendations address the scope creep problem:

- R3.1: Include a “buffer” – some surplus when estimating project resources (budget and schedule) – it allows (to some extent) adjusting to changes. It is especially advised when no prior experience can be used as a reliable estimation base.
- R3.2: Define project scope from the beginning. This task must involve key stakeholders and result in defining, agreeing, confirming and documenting high-level requirements: business goals, product vision, product scope and main constraints. Requirements which are known but will not be implemented within the current project should also be documented and assigned appropriate statuses.
- R3.3: Change request issued by the customer should be processed by the change control process. Its first step is the impact assessment. Apart from analyzing influence on other requirements, it should include: verification of compliance to business goals and estimation of required resources.
- R3.4: Use requirements priorities assigned by customer representatives to distinguish essential and secondary requirements.
- R3.5: Include requirements sign-off activity in a project (i.e. the customer should confirm in writing that documented requirements are valid).
- R3.6: Approved scope changes should be immediately communicated to all stakeholders.
- R3.7: Approved scope changes should result in adjusting the project’s budget and schedule.
- R3.8: Apply prototyping technique to prevent changes stemming from misunderstandings between stakeholders and developers (as mockups and other throw-away prototypes are relatively cheap to develop).
- R3.9: If impact analysis reveals change proposal to be harmful (e.g. impossible to implement within given constraints, not consistent with the business goals) – confront the customer with it.

D. P4 - Too short time for analysis

Reduced time available for analysis can be just a planning flaw, but often it is rather a result of the customer’s pressure

to produce software as soon as possible and reduce or skip all activities not directly resulting in code development. The customer can e.g. refuse to pay for RE/BA activities or insist on planning them very short and/or only at the very beginning of the project. It is very likely that such attitude would lead to misunderstandings, rework, delays, and increased costs, so it should be prevented by e.g.:

- R4.1: The most essential thing is to raise customer’s awareness. The customer often has a limited knowledge about software project organization and does not understand the importance of the RE/BA activities. Information about benefits of well conducted RE/BA activities as well as risks caused by reducing them should be provided to the customer.
- R4.2: It is important to build an atmosphere of trust in customer-supplier cooperation. The customer should be convinced that activities and techniques used in RE/BA are selected because of their contribution to the project success and not in order to increase costs.
- R4.3: Negative examples can be used to convince the customer e.g. cases where invalid requirements led to significant rework effort and related costs.

E. P5 - Stakeholders’ low availability

A stakeholder is likely to be busy with his/her everyday work duties and thus unable to contribute much time to the project e.g. to be interviewed by the analyst or to validate the specified requirements. Stakeholders are however indispensable for the acquisition of the domain knowledge as well as the elicitation and validation of the requirements. In order to win their commitment and to minimize their additional effort, the following practices can be considered:

- R5.1: At the beginning of the project present the customer with the RE/BA plan. The plan should define the activities and their phases as well as clearly indicate what kind of stakeholders’ involvement is required and when.
- R5.2: When the representatives of key stakeholders are already identified, inform the customer’s executives about the need to add tasks related to the participation in RE/BA to the representatives’ responsibilities and schedules.
- R5.3: When the information from human stakeholder is to be obtained, a direct communication or video-conference is preferred, as other communication means would probably prolong this process. An analyst should however strive to minimize disruptions to the customer organization’s processes and to the stakeholders’ tasks by appropriate planning of the meetings.
- R5.4: Each meeting between an analyst and the stakeholder(s) should have clearly assigned participants and goal (when it is achieved, the meeting ends). Also the maximum duration of the meeting should be planned ahead.

- R5.5: Number of the meeting participants should be limited because otherwise the discussion is difficult to control (because of subgroups emerging, some participants getting frustrated with topics they do not comprehend or find not interesting to them etc.).

F. P6 - Solutions issued instead of requirements

RE/BA should be focused on identifying requirements which are then processed to design the best solutions (within given constraints) that fulfil them. It is however quite typical that what a stakeholder describes is neither a requirement nor a constraint, but a particular solution (which may not be optimal at all, considering the bigger picture). The following recommendations were identified to deal with such problem:

- R6.1: When information obtained from a stakeholder turns out to be a solution instead of a requirement, it should not be ignored as it is a valuable lead for further inquiry. An analyst should note it down, assign it to a separate category (other than requirements) and use in further interviews as a means to steer the discussion.
- R6.2: A particular solution described by a stakeholder may indicate that he/she already knows a similar software (perhaps a legacy or competitor system). An analyst should study such system and review its features with stakeholders, trying to derive requirements. Missing requirements can be identified as a side effect.
- R6.3: Stakeholders often prefer to discuss solutions because they perceive it as something more concrete and easier to comprehend. It is important for an analyst to have a sufficient knowledge about problem domain (acquired earlier) and use it when interviewing a stakeholder to generalize discussed concept and express it as a requirement.
- R6.4: A feasible way to discover an actual requirement behind a design solution is to ask „why” questions i.e. request a stakeholder to provide the rationale behind that particular solution he/she describes.
- R6.5: To derive a requirement from a design solution, an analyst has to paraphrase and generalize the input received from a stakeholder. Applying a combination of requirement elicitation techniques is beneficial, because more diversified information can be obtained this way and provide a better basis for derivation of requirements.

G. P7 - Requirements in the form of change requests only

Requirements elicitation is usually a difficult task as stakeholders can have problems with articulating their needs or even realizing them. It leads to a situation when stakeholders “don’t know what they want, but know what they don’t want when presented with it” and issue change requests to the published release of a developed system. To prevent it, the following practices can be considered:

- R7.1: When human stakeholders turn out to be uncooperative, requirements can be elicited (to some extent) from other sources through e.g. document analysis - reviewing organizational charts, procedures, existing process models, memos etc.
- R7.2: The main scope of the developed system’s functionality and other essential features are probably included in the contract document, which can be used as a starting point for requirements elicitation.
- R7.3: If the system under development is to support existing or modified business processes in the customer organization, observation technique can be used to allow the analyst to gather the information firsthand by witnessing the work activities.
- R7.4: Existing IT systems used in the customer’s organization can be a valuable source of requirements. Moreover, discussing obsolete, missing and suboptimal functions of the existing system with the stakeholders can be much more effective than just asking them about their needs.
- R7.5: After identifying initial requirements, an analyst should facilitate a workshop to present and discuss them with a group of stakeholders. Despite being rather generic and not validated, such requirements can constitute the basis for discussions. It is essential to present the requirements in a communicative way e.g. by using prototypes. A facilitated discussion during which the stakeholders can refer to the initial requirements can enable them to communicate requirements they were unable to articulate earlier.

H. P8 - Conflicting requirements

Requirements elicitation is followed by requirements analysis which is likely to uncover issues like ambiguous, incomplete and, last but not least, conflicting requirements. Conflicting requirements may be attributed to mistakes, but more likely they are the result of different points of view, needs and expectations of various stakeholders and, as such, should be properly managed:

- R8.1: It is worth to identify an authorized customer’s representative, who would be capable of resolving conflicts in case stakeholders cannot do it themselves.
- R8.2: When conflicting requirements occur, the first step should be to verify their consistency with the business goals and project scope. The requirements that fail such a test can be revealed as unfounded and, as a result, the conflict may be solved.
- R8.3: An analyst needs to acquire domain knowledge to be able to determine the degree to which particular requirements contribute to the achievement of the business goals and to the project success. It can be used as a criterion to choose one option from the conflicting ones.
- R8.4: When conflicting requirements occur, this matter should be discussed at a meeting with all involved

stakeholders. The analyst should moderate the discussion and aim at explicitly presenting the trade-offs, suggesting ways of resolving conflicts and reaching the consensus.

- R8.5: Two or more conflicting requirements (concerning e.g. a function) can be replaced with a single one (more generic or more complex) that addresses the goals/rationales of each of them.
- R8.6: If substantial effort has been made and still stakeholders cannot (or are unwilling to) reach consensus, such issue can be delegated to a decisive entity e.g. a project sponsor or the senior management of the customer organization.

I. P9 - Stakeholders ignore business goals

Business goals defined by authorized customer representatives are the very reason behind customer's decision to acquire an IT system and start a software project. Also a primary indicator of the project success is whether the developed software allowed to achieve the business goals. If business goals are not precisely specified or even completely ignored, then there is no point of reference to determine which requirements contribute to the business benefits of the customer organization. It is thus advised that:

- R9.1: An analyst should consult key representatives of customer organization and insist on defining business goals before eliciting lower level requirements. Business goals (specified in precise and unambiguous way) should then be disseminated among all parties involved, both from supplier and customer sides.
- R9.2: Business goals should be maintained during the whole project schedule. If business goals change (due to e.g. market situation), all parties involved in software project should be notified and requirements need to be reviewed w.r.t. the new goals.
- R9.3: Even if substantial effort was made to identify business goals, it is possible that some goals were omitted. When dealing with a requirement that cannot be associated with any goal, a possibility that the set of goals is incomplete should be considered. It may lead to defining an additional goal.

J. P10 - Requirements exceeding project's scope

When project scope is agreed between the customer and the supplier, usually the associated project constraints (budget, schedule) are defined in accordance with the scope. A situation that stakeholders define requirements exceeding this scope is dangerous, because it will likely also result in project not meeting its constraints. It may however be caused by invalid scope assumed at the beginning of the project. To avoid the problem, the following actions can be taken:

- R10.1: It is important to ensure that the definition of the project scope involves several stakeholders, selected on the basis of their domain knowledge and authority. It should prevent invalid scope definition.

- R10.2: Change management process needs to be defined and followed. Such process should treat the cases of requirements exceeding the scope as changes to the project scope. Impact analysis should be conducted for such changes and include the analysis of the change's influence on schedule and budget.
- R10.3: The agreed project scope should be documented and formally approved by an authorized customer representatives. An analyst can refer to such a document when particular stakeholders insist on adding requirements which surpass that scope.

V. INTERVIEWS WITH EXPERTS

After collecting the recommendations, the next step was to assess their effectiveness. This assessment was supposed to be based on real-life experiences, not speculations. We decided to interview experts for this purpose. We selected 3 analysts whom we considered to be experts in RE/BA domain, with regard to their knowledge (confirmed by trainings completed and certificates obtained) and professional experience (work history including several companies and projects from various business domains). Their brief characteristics are given below:

- Expert A – 6 years of experience as an analyst (11 years in IT in total). Software projects from the following business domains: transportation, logistics, medical, financial. PhD in computer science, REQB FL certificate holder.
- Expert B – 8 years of experience as an analyst (10 years in IT in total). Software projects from the following business domains: public administration, telecommunications, transportation. IIBA CBAP certificate holder. Active participation in RE/BA professional association (local chapter board member).
- Expert C – 15 years of experience as an analyst. Software projects from the following business domains: insurance, public administration, government, electronics and telemetry. REQB FL certificate holder. Job history including middle- and high-level management positions and the role of coach in several commercial RE/BA courses.

Each expert's task was to review the presented recommendations and provide his/her feedback. For each problem, an expert had to answer the following questions:

- Have you encountered this problem in your work experience as an analyst?
- Which of the listed recommendations have you tried to address this problem?
- Which among the applied recommendations turned out to be effective?
- Which among the applied recommendations failed to mitigate the problem?

TABLE II.
ASSESSMENT RESULTS FOR RECOMMENDATIONS TO PROBLEMS P1-P5

Problem	Recommendation	A	B	C	Remarks
P1	R1.1	Y	U	Y	A: The recommendations proved effective but only when used altogether and not selectively e.g. the identification of the business goals only. B: The communication and education of the customer is the key (none of the recommendations would succeed if used without it). It is also useful to define the Minimum Viable Product and other versions with enhanced functionality, estimate the cost of each version. C: If possible, try to simplify the solution (e.g. scope of functionality) in a way not compromising the business goals.
	R1.2	Y	U	Y	
	R1.3	Y	U	X	
	R1.4	Y	U	U	
	R1.5	Y	U	Y	
	R1.6	Y	U	N	
	R1.7	U	U	N	
	R1.8	U	U	X	
P2	R2.1	U	Y	X	A: Interviews (unstructured, including apparently obvious matters) and workshops are the most effective elicitation techniques to address this problem. B: R2.1 is a reasonable recommendation in general, but an analyst should be aware that sometimes it may fail (a specific customer deviating from standard processes known in problem domain, an existing system which is poorly tailored to the needs). B: R2.5 requires good understanding of the template contents - if a template is used with an attitude to leave no section empty and there is no focus on quality of the content, then nothing good comes out of it. B: An additional way to address this problem is the prototyping which often reveals missing "obvious" requirements. C: It is the requirements validation rather than the requirements analysis (R2.4) that can reveal hidden requirements. C: Additional recommendations: 1. Requirements validation through a dedicated meeting; 2. Business process modelling with mapping between the processes and the requirements.
	R2.2	U	Y	Y	
	R2.3	Y	U	Y	
	R2.4	Y	Y	Y	
	R2.5	U	U	X	
	R2.6	U	Y	N	
P3	R3.1	U	U	N	A: Minimum Viable Product concept is also useful here. B: Presented recommendations are good ideas, but none of them guarantees the scope creep prevention e.g. a buffer (R3.1) may be insufficient, some customer representatives do not feel obliged by sign-off and reject earlier agreements etc. B: Other recommendations could be: 1. accept small changes (up to a specific effort estimation) and move bigger ones to future releases; 2. develop iteratively in time & material mode; 3. acquire knowledge about problem domain and customer organization (structure, processes) to get better understanding of the possible scope definitions and avoid surprises.
	R3.2	Y	U	Y	
	R3.3	Y	U	Y	
	R3.4	Y	U	X	
	R3.5	U	U	N	
	R3.6	U	U	X	
	R3.7	U	U	X	
	R3.8	U	U	Y	
	R3.9	U	U	Y	
P4	R4.1	U	U	Y	B: All 3 recommendations are good ideas but some customers still just say "no" and refuse to listen to any arguments B: Other ideas: 1. make project phases/activities non-negotiable, the project should be "sold" to the customer only as a whole; 2. develop iteratively and demonstrate the results (analysis will be divided into many smaller tasks spread over time and the customer will get partial result earlier); 3. try to negotiate stakeholders' commitment e.g. analysis will be shorter if a given stakeholder joins the development team for a week. C: It is crucial to develop the RE/BA plan (consistent with overall project plan and customer's expectations) which should specify who is to be involved from customer's side, in which activities and why such involvement is necessary.
	R4.2	Y	U	Y	
	R4.3	U	U	N	
P5	R5.1	N	U	Y	A: Stakeholders' tasks and expected input have to be precisely defined. B: Again, recommendations are worth trying, but if the low availability is not due to the lack of awareness but e.g. heavy workload then none of suggested actions will change that. B: Additional recommendations: 1. Escalation - ensure stakeholders' availability by talking to their superiors (executives, project sponsor etc.); 2. Include the specific clauses about maximum response time and minimal effort required from customer's side in the project contract.
	R5.2	U	U	X	
	R5.3	Y	U	Y	
	R5.4	Y	U	Y	
	R5.5	U	U	Y	

The summarized results of the assessments are presented in Tables II and III. Both tables have the same columns: problem ID; associated recommendations IDs; assessments by experts A, B and C; additional remarks made by experts.

The following symbols are used in the table to denote the assessment results:

- X – Recommendation was not used by an expert to mitigate a given problem;

- Y – (Yes), recommendation was used and proved effective;
- N – (No), recommendation was used and failed to mitigate the problem;
- U – (Uncertain), recommendation was used but it is not possible to determine its exact contribution to the problem-solving or the outcome of applying the recommendation differed from project to project;

In addition, the colors are used to distinguish the assessment results. Only expert A denied experiencing some of the enumerated problems (P7, P9 and P10), thus X values are used for all associated recommendations.

TABLE III.
ASSESSMENT RESULTS FOR RECOMMENDATIONS TO PROBLEMS P6-P10

Problem	Recommendation	A	B	C	Remarks
P6	R6.1	U	U	N	A: It is not recommended to review an existing system with a stakeholder, because it results in closing his/her mind to the options other than present in that system. B: Some people are stubborn and resistant to any reasoning. They will most likely insist on a particular solution even when presented with strong arguments against it. C: Education of stakeholders, so they know what a requirement is and how to express it - such issue should be brought to the main customer's representative.
	R6.2	N	U	Y	
	R6.3	U	U	Y	
	R6.4	U	U	N	
	R6.5	Y	U	N	
P7	R7.1	X	U	Y	B: R7.2 does not make much sense - if stakeholders cannot articulate their needs, how to determine the scope/features to be written in the contract? B: Alternative sources of requirements (documents, observation, existing systems) may not provide all the necessary information or be unavailable (e.g. no existing IT system used, new business processes not implemented yet and impossible to observe). B: Prototyping or incremental development can be effective as the cost of changes is lower. C: Additional recommendations: 1. Business process modelling with mapping between the processes and the requirements; 2. Specify the stakeholders' points of view.
	R7.2	X	N	N	
	R7.3	X	U	N	
	R7.4	X	U	Y	
	R7.5	X	Y	Y	
P8	R8.1	X	U	X	B: Relying on an authorized representative is a good idea, provided that such person is willing to take responsibility and make difficult decisions. B: Some stakeholders may insist on "their" requirements regardless of the consistency with the business goals. B: Additional recommendation: use multi-criteria assessment (including priority, difficulty, cost, conformance to standards etc.) and apply it to the conflicting requirements to choose the optimal one (w.r.t. those criteria).
	R8.2	X	U	Y	
	R8.3	X	U	N	
	R8.4	Y	Y	Y	
	R8.5	Y	U	X	
	R8.6	Y	U	Y	
P9	R9.1	X	U	Y	B: If stakeholders refuse to define the goals, no one can force them or do it on their behalf. B: Additional recommendations: 1. Educate the stakeholders about the importance of business goals and the risk of ignoring them; 2. Try to define business goals on the basis of elicited requirements and ask the stakeholders to validate them (but be aware of a risk that they will confirm the goals without actually considering them). C: Specify the stakeholders' points of view.
	R9.2	X	U	Y	
	R9.3	X	U	N	
P10	R10.1	X	U	N	B: Additional recommendation: A workshop during which the stakeholders associate their requirements to business goals/high-level requirements. C: Specify the stakeholders' points of view. Constantly monitor the scope.
	R10.2	X	Y	Y	
	R10.3	X	U	Y	

VI. DISCUSSION

A. Discussion of results

Recommendations found cover all the problems we intended to address and there was no single case that a given problem was not recognized by literature or without any suggestions how to cope with it. The list of recommendations we assembled is however far from being complete, as analysts we interviewed reported several additional remedies to these problems from their experience. Keeping in mind that it was a very small scale study (3 interviewees), we

should expect more recommendations if a larger group of analysts were involved.

The recommendations differ according to their nature: some are just choices of a particular technique (for e.g. requirements elicitation), some are rather related to the processes and organizational issues (e.g. planning/monitoring activities, participation of particular people), while other concern cooperation and relationships (e.g. educating stakeholders, atmosphere of trust). Quite often, multiple recommendations benefit from being used together.

Assessments of the recommendations' effectiveness collected from the interviews are inconclusive at best.

Knowing the complexity of the RE/BA processes (and the software development in general) as well as the differences between the organizations, projects and business domains, we did not expect the same answers from each interviewee. However, differences in their assessments are greater than expected - there are literally two cases in which all 3 experts made the same assessment. To some extent it can be attributed to differences in experts' perception/attitude. It is clearly seen that expert B avoided giving definite answers (very few Y or N, mostly U assessments). This person was however the most active in providing additional remarks about e.g. factors influencing recommendations' effectiveness or situations that they would not work. The remaining two experts were more willing to summarize their experiences in a yes/no answer.

We considered this interview study as an initial validation of the recommendations found. Results of the study clearly indicate that it is better to be careful with using recommendations, as the outcome can differ in various domains, projects and other settings. Thus, such a list as the one assembled by us has a limited use, as it is neither complete nor provides enough guidelines about the context a given recommendation should or should not be used. We see two possible improvements. The first is to extend the list with additional recommendations identified through the literature reviews, interviews, surveys etc. Such list can grow and become more difficult to use, but it seems necessary as apparently no problem (at least not those listed in Table I) can be mitigated by a single action with a 100% effectiveness. The second improvement is to annotate the recommendations with guidelines about their applicability, limitations, trade-offs etc. (perhaps in a similar way as for the design patterns).

Another issue worth addressing in future is to consider problems and recommendations in a wider context. RE/BA does not exist in isolation but is a part of an overall software development process and strongly influences e.g.: software architecture [25] or testing (also verification & validation in general) [26]. When developing and presenting recommendations, it would be worthwhile to take into account contextual factors and consequences to software project activities and artefacts other than just those directly related to RE/BA.

B. Discussion of validity

We are aware of several limitations of our study. The number of interviewees is the most important one, as small scale studies cannot be considered as very convincing. A larger group of experts would make results more valid.

Another issue is the competence and representativeness of the interviewees. Knowledge and experience were applied as criteria for interviewee selection. Moreover, the diversity of projects and business domains each of them worked for is an argument for their credibility. We did not analyze their representativeness among the general population of analysts

though, which is a possible validity threat, especially if results were to be generalized as applicable worldwide.

The quality of input data should be considered. In this case, such input was the list of recommendations presented to interviewees. Although an effort was made to assemble it on the basis of multiple sources, it certainly was not exhaustive, but that was not necessary – interviewees were only supposed to assess each of the presented recommendations and optionally report others known to them.

A common validity threat to interview-based research is the honesty of interviewees. We mitigated this threat by arranging interviews with volunteers only and assuring that they would remain anonymous. Subjectivity is also an inevitable aspect of interviews – despite our instructions to give assessments on experience basis, no strict, quantitative criteria were defined and interviewees could differ in their judgements. For example, a recommendation which was successfully applied in many cases but proved unsuccessful once could be classified as effective (Y) by one expert and uncertain (U) by another.

Finally, researchers can be the source of validity threats too by their possible bias regarding e.g. conviction about the effectiveness of particular recommendations. We addressed the first issue by providing interviewees problems and recommendations in a written form, without any additional suggestions and by interpreting the outcome separately by 2 researchers and then comparing their opinions.

VII. CONCLUSION

In this paper we reported a research study aimed at identifying and assessing known recommendations that address top requirements-related problems in the IT industry in Poland. Answers to the research questions formulated in the introductory section were obtained through reviewing relevant sources (RQ1) and interviewing experienced analysts (RQ2). These answers were presented in Section IV – Recommendations (RQ1) and in Tables II-III (RQ2).

The insight gathered from the study can be briefly summarized as follows. There are many recommendations available and described in various sources. They can be considered accurate in general – there was not a case of our experts rejecting a recommendation unanimously as inapplicable or invalid (maybe with the exception of R7.2). Recommendations are, however, not the “silver bullets” that can be used in any context with a guarantee of success. Experience-based assessments made by a small number of experts proved to be very diverse, which, on the basis of the interviews, we attribute to the contextual differences (domain, customer and supplier profiles, project organization etc.). Such applicability context is not necessarily specified as part of the documented recommendations.

The possible future research include: identifying a larger set of recommendations (by e.g. literature reviews, surveys, interviews), specifying their applicability context and other constraints (if such issues are not provided together with a

recommendation itself, they can be established by surveying/ interviewing practitioners), and making additional assessments of recommendations' effectiveness (in a more specific context, by a larger group of experts). The possible outcome is a sort of "catalogue" of recommendations that practitioners could review and select recommendations to apply in their project as a response to encountered problems.

ACKNOWLEDGMENT

We are grateful to the interviewed experts for their participation and sharing their experiences.

REFERENCES

- [1] R. N. Charette, "Why Software Fails", *IEEE Spectrum*, vol. 42, no. 9, pp. 42–49, 2005, <https://doi.org/10.1109/mspec.2005.1502528>
- [2] J. McManus and T. Wood-Harper, "Understanding the Sources of Information Systems Project Failure - A study in IS project failure", *Manag. Serv.*, vol. 51, no. 3, pp. 38–43, 2007.
- [3] D. Mendez Fernández et al., "Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice", *Empir. Softw. Eng.*, vol. 22, no. 5, pp. 2298–2338, 2017, <https://doi.org/10.1007/s10664-016-9451-7>
- [4] B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering", *Proceeding FOSE '07 2007 Futur. Softw. Eng.*, pp. 285–303, 2007. <https://doi.org/10.1109/fose.2007.17>
- [5] A. Przybyłek, "A Business-oriented Approach to Requirements Elicitation", in *9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'14)*, Lisbon, Portugal, pp. 152–163, 2014. <https://doi.org/10.5220/0004887701520163>
- [6] A. Przybyłek and M. Zakrzewski, "Adopting Collaborative Games into Agile Requirements Engineering", in *13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'18)*, Funchal, Madeira, Portugal, pp. 54–64, 2018, <https://doi.org/10.5220/0006681900540064>
- [7] T. Hall, S. Beecham and A. Rainer, "Requirements problems in twelve software companies: an empirical analysis", *IEE Proc. - Softw.*, vol. 149, no. 5, p. 153, 2002, <https://doi.org/10.1049/ip-sen:20020694>
- [8] N. K. Sethia and A. S. Pillai, "A study on the software requirements elicitation issues - its causes and effects", *2013 Third World Congr. Inf. Commun. Technol. (WICT 2013)*, pp. 245–252, 2013, <https://doi.org/10.1109/wict.2013.7113143>
- [9] D. Mendez Fernandez et al., "Naming the Pain in Requirements Engineering: Comparing Practices in Brazil and Germany", *IEEE Softw.*, vol. 32, no. 5, pp. 16–23, 2015, <https://doi.org/10.1109/ms.2015.122>
- [10] D. Firesmith, "Common requirements problems, their negative consequences, and the industry best practices to help solve them", *J. Object Technol.*, vol. 6, no. 1, pp. 17–33, 2007, <https://doi.org/10.5381/jot.2007.6.1.c2>
- [11] B. Davey and K. Parker, "Requirements elicitation problems: a literature analysis", *Issues Informing Sci. Inf. Technol.*, vol. 12, pp. 71–82, 2015, <https://doi.org/10.28945/2137>
- [12] A. Jarzębowicz and W. Ślesiński, "What is Troubling IT Analysts? A Survey Report from Poland on Requirements-related Problems", in *Proc. of 20th KKIO Software Engineering Conference*, Advances in Intelligent Systems and Computing vol. 830, pp. 3–19, Springer, 2018, https://doi.org/10.1007/978-3-319-99617-2_1
- [13] K. el Emam and N. H. Madhavji, "A field study of requirements engineering practices in information systems development", *International Conference on Requirements Engineering*, pp. 68–80, 1995, <https://doi.org/10.1109/isre.1995.512547>
- [14] E. Bjarnason, K. Wnuk, and B. Regnell, "A case study on benefits and side-effects of agile practices in large-scale requirements engineering", *Proc. 1st Work. Agil. Requir. Eng. - AREW '11*, pp. 1–5, 2011, <https://doi.org/10.1145/2068783.2068786>
- [15] F. G. De Oliveira Neto, J. Horkoff, E. Knauss, R. Kasauli, and G. Liebel, "Challenges of aligning requirements engineering and system testing in large-scale agile: A multiple case study" *Proc. 2017 IEEE 25th Int. Requir. Eng. Conf. Work. REW 2017*, p.p. 315–322, 2017, <https://doi.org/10.1109/rew.2017.33>
- [16] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed Agile: An empirical study", in *Proc. of 24th Requirements Engineering: Foundation for Software Quality Conference*, pp. 139–154, 2018, https://doi.org/10.1007/978-3-319-77243-1_9
- [17] A. M. Hickey and A. M. Davis, "Elicitation technique selection: How do experts do it?", in *Proceedings of the IEEE International Conference on Requirements Engineering*, 2003, pp. 169–178. <https://doi.org/10.1109/icre.2003.1232748>
- [18] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study", *IEEE Softw.*, vol. 25, no. 1, pp. 60–67, 2008, <https://doi.org/10.1109/ms.2008.1>
- [19] D. Paul and L. Y. Tan, "An Investigation Of The Role Of Business Analyst In IS Development", *ECIS 2015 Proc.*, pp. 1–14, 2015.
- [20] K. Wiegers and J. Beatty, "*Software Requirements*", 3rd ed. Microsoft Press, 2013, ISBN: 978-0735679665.
- [21] D. Leffingwell and D. Widrig, *Managing Software Requirements*, Pearson Education, 2003, ISBN:032112247X
- [22] B. Chrabski and K. Zmitrowicz, *Requirements Engineering in Practice* (in Polish: Inżynieria Wymagań w Praktyce), Wydawnictwo Naukowe PWN, 2015, ISBN: 9788301180188
- [23] M. Bartyzel, *Tailored software - how to speak to customers who don't know what they want* (in Polish: Oprogramowanie szyte na miarę. Jak rozmawiać z klientem, który nie wie, czego chce). Wydawnictwo Helion, 2012, ISBN: 978-83-246-3932-8
- [24] E. Bernroider and M. Ivanov, "IT project management control and the Control Objectives for IT and related Technology (CobiT) framework", *Int. J. Proj. Manag.*, vol. 29 no. 3, pp. 325–336, 2011, <https://doi.org/10.1016/j.ijproman.2010.03.002>
- [25] J. Cleland-Huang, R. S. Hanmer, S. Supakkul, and M. Mirakhori, "The Twin Peaks of Requirements and Architecture", *IEEE Softw.*, vol. 30, no. 2, pp.24–29, 2013, <https://doi.org/10.1109/MS.2013.39>
- [26] E. Bjarnason et al., "Challenges and practices in aligning requirements with verification and validation: a case study of six companies", *Empir. Softw. Eng.*, vol. 19, no. 6, pp. 1809–1855, 2014, <https://doi.org/10.1007/s10664-013-9263-y>