Interview Review: Detecting Latent Ambiguities to Improve the Requirements Elicitation Process

Alessio Ferrari*, Paola Spoletini†, Beatrice Donati‡, Didar Zowghi§ and Stefania Gnesi*
* CNR-ISTI, Pisa, Italy, Email: alessio.ferrari@isti.cnr.it, stefania.gnesi@isti.cnr.it
†Kennesaw State University, Marietta (GA), USA, Email: pspoleti@kennesaw.edu
‡University of Florence, DILEF, Florence, Italy, Email: beatrice.donati@unifi.it
§Faculty of Engineering and Information Technology, UTS, Sydney, Australia, Email: didar.zowghi@uts.edu.au

***Abstract*—In requirements elicitation interviews, ambiguities identified by analysts can help to disclose the tacit knowledge of customers. Indeed, ambiguities might reveal implicit or hard to express information that needs to be elicited. The perception of ambiguity might depend on the subject who is acting as analyst, and different analysts might identify different ambiguities in the same interview. Based on this intuition, we propose to investigate the difference between ambiguities explicitly revealed by an analyst during a requirements elicitation interview, and ambiguities annotated by a *reviewer* who listens to the interview recording, with the objective of defining a method for interview review. We performed an exploratory study in which two subjects listened to a set of customer-analyst interviews. Only in 26% of the cases the ambiguities revealed by the analysts matched with the ambiguities found by the reviewers. In 46% of the cases, ambiguities were found by the reviewers, and were not detected by the analysts. Based on these preliminary findings, we are currently performing a controlled experiment with students of two universities, which will be followed by a real-world case study with companies. This paper discusses the current results, together with our research plan.**

I. Introduction

The review of software process artifacts, which include requirements as well as source code [1], is an effective practice to improve the quality of products [2]–[5]. In particular, the benefits of requirements reviews have been highlighted by several studies, especially for what concerns the identification of defects in requirements specifications [3], [6], [7]. Nevertheless, despite the usage of requirements reviews dates back at least 40 years [6], challenges exist for their widespread application in the software industry [8], [9]. Among the challenges, Salger highlights that *"Software requirements are based on flawed 'upstream' requirements and reviews on requirements specifications are thus in vain"* [8]. This observation poses an emphasis on the need to ameliorate early requirements elicitation activities, especially to improve the *completeness* of the specifications, a quality attribute that is recognised to be hard to assess by means of reviews [10].

Requirements elicitation activities often start with an interview between a customer and a requirements analyst [11]–[14]. One of the aspects that greatly affects the success of interviews, and in turn the completeness of the gathered requirements, is *tacit knowledge* [15]. Tacit knowledge is system relevant information that remains unexpressed during requirements elicitation, and it is therefore not transferred to the specification. Techniques were developed to facilitate the disclosure of tacit knowledge [16]–[19], and, among these techniques, the detection of ambiguity has been highlighted as a powerful tool [17]. Indeed, when an ambiguity is detected in the words of a customer *during* an interview, this can lead to the identification of unexpressed, system-relevant aspects that need to be made explicit [20]. However, tacit knowledge remains an open problem in requirements engineering [16], and further techniques are required to elicit latent needs and domain information from customers.

The idea presented in this paper is to leverage ambiguity to disclose tacit knowledge *after* the interview. Our proposal is to apply reviews to interview recordings, thus bringing the review technique upstream in the software process, to address the problem of requirements completeness. The rationale behind the proposal is that ambiguities in the words of a customer can be perceived in different ways by different analysts, as observed for ambiguities in written requirements [21], [22]. The idea of the method is particularly simple, in line with the RE'17 theme *"Desperately Seeking Less": The Role of Simplicity and Complementarity in Requirements*. In the proposed method, the analyst performs the interview with the customer, and records the audio of the dialogue. The audio is reviewed by a reviewer, who annotates the ambiguities perceived, and lists the questions that he would have asked if he had been the analyst. The questions are used for further clarifications in future interactions with the customer. To provide an empirically proven method, our research is conducted according to a three phase experimental process, during which the method is defined, consolidated and incrementally assessed. Specifically, the phases are: (1) an exploratory study; (2) a controlled experiment with two independent groups of students from University of Technology Sydney (UTS) and Kennesaw State University (KSU); (3) an industrial case-study. We completed the first phase of this plan, and we are conducting the controlled experiment, while contacts with companies have been established to perform the last phase.

The remainder of the paper is structured as follows. In Sect. II, we summarise related works concerning ambiguity in RE, and review techniques. In Sect. III, our research plan is outlined. In Sect. IV, V and VI we describe the different experimental phases of our research. In Sect. VII, we outline its potential risks. In Sect. VIII, we provide final remarks.

2332-6441/17 $31.00 © 2017 IEEE
DOI 10.1109/RE.2017.15

II. Related Work

A. Ambiguity in Requirements

Ambiguity in natural language has been studied extensively in RE, especially in relation to its occurrence in written requirements. In particular, strategies were defined to *prevent* ambiguities by means of formal approaches [23]–[25] or constrained natural languages [26]–[28]. The works of Kof [23], and tools like Circe-Cico [24] and LOLITA [25], which transform requirements into formal/semi-formal models, have ambiguity prevention among their objectives. Concerning the use of constrained natural languages, the EARS [26] and the Rupp's template [27] are well known constrained formats for editing requirements. Arora *et al.* [28] defined an approach to check the conformance of requirements to these templates.

Other approaches aim to *detect* ambiguities in requirements. Most of these works stem from the typically defective terms and constructions classified in the ambiguity handbook of Berry *et al.* [29]. Based on these studies, tools such as QuARS [30], SREE [31] and the tool of Gleich *et al.* [32] were developed. More recently, industrial applications of these approaches were studied by Femmer *et al.* [33] and by Rosadini *et al.* [21]. As shown also in these studies, rule-based approaches tend to produce a high number of false positive cases – i.e., linguistic ambiguities that have one single reading in practice. Hence, *statistical* approaches were proposed by Chantree *et al.* [34] and by Yang *et al.* [35] to reduce the number of false positive cases, referred as *innocuous ambiguities*.

All these works focus on written requirements, and, with the exception of Chantree *et al.* [34] and Yang *et al.* [35], they focus on the *objective* facet of ambiguity, assuming that the majority of the cases could be identified by focusing on a set of typically dangerous expressions. Our previous studies [17], [20] showed that these expressions account only for a limited number of ambiguities that occur in requirements elicitation interviews, in which the *subjective* and contextual facets become dominant. These facets are actually those that we wish to leverage in the current work.

B. Requirements Review

The review of software products is the subject of the IEEE Std 1028-2008 [1], which discusses five types of reviews, namely management reviews, technical reviews, inspections, walk-throughs and audits. In this paper, we focus on the *inspection* type, which is a *systematic peer-examination that [...] verifies that the software product exhibits specified quality attributes [...]* and *collects software engineering data*. Early and successful techniques for requirements inspection were provided by Fagan [6] and Shull *et al.* [3], while a survey on the topic was published by Arum *et al.* [36]. Katasonov and Sakkinen [37] provide a categorisation for reading techniques to be applied in inspection reviews, distinguishing between ad-hoc, checklist-based, defect-based, perspective-based, scenario-based and pattern-based. The technique proposed in our paper is *defect-based*, since it focuses on a particular type of defect, namely ambiguity. More recent works on requirements review are those by Salger [8] and by Femmer *et al.* [9]. Both the works list the *challenges* that requirements review faces in practice, which go from the long time required for its implementation [9], to the need to have more effective elicitation techniques [8]. This latter goal is pursued by Karras *et al.* [38], who developed a tool for video inspection of requirements workshops. The majority of the cited studies focus on reviews applied to specifications, while propose to analyse interviews. Our work differs also from that of Karras *et al.* [38], since we suggest to analyse only the audio of interviews, and we focus on ambiguity, a communication defect that is not considered by this previous study.

III. Proposed Research

The goal of our study is to provide an empirically validated method for interview review. Therefore, we conduct an empirical research, in which the method is defined and assessed. The hypothesis for our research is as follows:

Hypothesis: *Review of requirements elicitation interviews allows identifying ambiguities that can be leveraged to ask useful follow-up questions in future interviews.*

From the hypothesis, we identified three research questions:

RQ1: Is there a difference between ambiguities explicitly revealed by an analyst during an interview, and ambiguities identified by a reviewer who listens to the interview recording?

RQ2: Is there a difference between ambiguities identified by the analyst when listening to the interview recording, and ambiguities identified by a reviewer who listens to the interview recording?

RQ3: Can the ambiguities identified during interview review be used to ask *useful* questions in future interviews?

Each question is answered by means of empirical studies with different degrees of rigor and required effort. For RQ1, we conducted an exploratory study in which a set of interviews performed by students in a role playing context was reviewed for ambiguities by two researchers. To consolidate the answer to RQ1, and answer RQ2, we have planned a controlled experiment, in which students will act as analysts and reviewers. The controlled experiment will be replicated in two universities, namely KSU and UTS. For RQ3, we have planned a real-world case study with companies to assess the actual utility of the method. A critical analysis of each study produces input for setting up the following one, and, in particular, for refining the review method. In the next sections, we describe the current status of the different studies.

IV. RQ1: Exploratory Study

The goal of the exploratory study is to understand whether the basic intuition of the proposed method – i.e., the idea that different ambiguities may emerge, when an interview is listened by different subjects – is actually grounded. To this end, we define a preliminary version of the review method, and two authors of the current work apply it on a set of interviews performed by students.

TABLE I: Excerpt of a spreadsheet of reviewer 1.

Fragment	Time	D	Type	Question
I want an app in which the people can log into the system	00:10	B	mul und	A: Which kind of platform would you use? R: Is it an application for mobile, is it a Web app, or something else?
I'm gonna put a text into a field, I'm gonna set a time, I'm gonna set the recipient, and it's gonna text that person at that time	00:30	A	-	A: Why would you need that?
I can do quick text as well	08:02	R	int unc	R: What is quick text?

A. Study Design

The study consists of two phases: first, in which interviews are performed, and second, in which interviews are reviewed.

a) Interview: A set of role-playing interviews was performed by 38 students of KSU. The recruited students belonged to User-Centered Design course, composed of undergraduate students of the 3rd and 4th year. The students were divided into 2 groups, namely analysts and customers. The customers were required to think about a novel computer intensive system that they would like to be developed, and were given a week to think about the product. The analysts were provided with a two hours lecture on requirements elicitation interviews delivered by the 2nd author, in which they received an introduction on different types of interviews. The class uses a reference book [39] and additional lecture notes. The interviews took place simultaneously at KSU, and the time slot allocated was 20 minutes. The students conducted *unstructured* interviews [14]. This was considered as the most suitable approach in this context, in which analysts are exploring ideas for new products (e.g., mobile/desktop apps, games) for which they have no background information.

b) Review: Among the 19 interview recordings, a random sample of 10 recordings was chosen for review. The reviewers are the 1st and 3rd authors of the current paper, a researcher in requirements elicitation, and a professional analyst, respectively. The two reviewers – identified as reviewer 1 and 2 – were required to independently listen to each interview and to report ambiguous situations in a spreadsheet, according to the following protocol:

1) For each interview, add a line to the spreadsheet with the title of the audio file, and start listening the file.
2) Stop the audio in the following two cases: (a) you perceive an ambiguity in the words of the customer; (b) a question of the analyst denotes that he perceived an ambiguity in the words of the customer.
3) Whenever you stop the audio for the listed cases, add a line to the spreadsheet with the following content:

- **Fragment:** the fragment of speech of the customer that triggered the ambiguity;
- **Time:** the moment of the interview in which the customer produces the fragment;
- **Detector:** the actor that perceived the ambiguity, namely only you (write R, i.e., the reviewer), only the analyst (A), or both (B);
- **Type:** if the actor that perceived the ambiguity was R or B, specify the type of ambiguity you perceived according to the classification of Ferrari *et al.* [20], and using the following guidelines:
  - *interpretation unclarity* (**int unc**): you do not understand the fragment of the customer's speech;
  - *acceptance unclarity* (**acc unc**): you understand the fragment of the customer, and you have no reason to doubt that your understanding matches with the intended meaning of the customer. However, what you hear does not make sense to you. With the expression *does not make sense*, we intend that the fragment appears *incomplete* to you, or it has some form of *inconsistency* with what you have understood, or with your knowledge.
  - *multiple understanding* (**mul und**): you can give multiple interpretations to the fragment of the customer, and each interpretation makes sense to you.
  - *detected incorrect disambiguation* (**det inc dis**): you perceived an acceptance unclarity, and, later in the interview, you understand that your interpretation was not correct (i.e., it did not match with the intended meaning of the customer). In this case, you should change the label from "acc unc" to "det inc dis".
  - *undetected incorrect disambiguation* (**und inc dis**): you did not perceive an acceptance unclarity, but, at a certain point of the interview, you understand that your interpretation of a certain fragment of the customer was not correct. In this case, you should search for the fragment in the audio file, and add a line with the fragment.
- **Question:** if the ambiguity was perceived by you (R), report in the sheet the question that you would ask to the customer to clarify; if the ambiguity was perceived by the analyst, report the question asked by the analyst (A).

An excerpt of the spreadsheet of reviewer 1 is reported in Table I. The reviewers were also asked to keep track of the time employed to review each interview.

B. Results and Evaluation

To evaluate the results of this study, we went through the spreadsheets of the two reviewers, and computed (1) the number of ambiguities explicitly detected by the analyst (identified as a); (2) the number of ambiguities that were common between the reviewers and the analyst (b); (3) the number of ambiguities identified only by the reviewers (r). To evaluate these numbers, we consider the two reviewers as a *single* reviewer – separate statistics will be provided later in this paragraph. In particular, let A_1, B_1, R_1 the items in the spreadsheets of reviewer 1, in which the field "Detector" was marked as A, B and R, respectively. Let A_2, B_2, R_2 the corresponding items for reviewer 2. We have: $a = |A_1 \cup A_2| = 23$;

$b = |B_1 \cup B_2| = 21$; $r = |R_1 \cup R_2| = 38$. The total number of ambiguities identified in the whole process is given by $n = a + b + r = 82$. From these numbers, we see that the reviewers altogether were able to identify 46% of the ambiguities, the analyst alone identified 28% of the cases and the remaining 26% of the cases were common between the reviewers and the analyst. The agreement among the reviewers and the analysts, measured with Cohen's Kappa [40], is $k = -0.54$, indicating *disagreement* among reviewers and analyst ($k < 0$ indicates disagreement). These results indicate that a numerical difference exists in the ambiguities explicitly revealed by analysts and those detected by reviewers.

If we consider the reviewers separately, we have $|A_1| = 21, |B_1| = 14, |R_1| = 27$ and $|A_2| = 7, |B_2| = 15, |R_2| = 18$. From these numbers, we see that even a single reviewer can provide a relevant support for spotting out latent ambiguities. If we compute the agreement among the individual reviewers by comparing their fragments with "Detector" A, B and R, we obtain indications of disagreement, $k_A = -0.18, k_B = -0.45, k_R = -0.59$. This indicates that (a) reviewers disagree on which ambiguities were explicitly revealed by the analyst ($k_A, k_B < 0$), and (b) reviewers identified different ambiguities ($k_B, k_R < 0$). Further analysis on the types of ambiguities, to be performed in the subsequent empirical phases, will explain the *qualitative* differences between the ambiguities revealed, and the factors (e.g., domain knowledge [41]) influencing the disagreement.

Another aspect that we evaluated was the *time* employed by the reviewers for their task, with respect to the duration of the interviews. For a total of 2 hours and 37 minutes of recordings, reviewer 1 employed 5 hours, while reviewer 2 employed 8 hours and 33 minutes. The time required for interview review is therefore two to three times greater (2.75 times, in average) than the time of the interview.

C. Limitations and Input for the Controlled Experiment

We acknowledge a set of limitations of this exploratory study, which allows us to provide a more appropriate design for the subsequent phases:

L1. Population, Experience and Bias: our study used solely two reviewers, who are not a representative set of the population of reviewers. In addition, the gap in terms of experience between analysts and reviewers is consistent, and this, together with the researcher bias, may have led to the large discrepancies that we have seen between the number of detected ambiguities. To mitigate these aspects, we perform a controlled experiment in which two groups of students of different universities are involved as reviewers, while the researchers will perform solely the analysis of the data.

L2. Realism: a large part of the reviewed interviews did not appear totally realistic. Except for some cases in which the analysts were playing their role with sufficient participation, in many cases the analysts did not question sufficiently the statements of their customers, probably because the analysts were not evaluated by the instructor based on their interviews. Furthermore, customers often proposed ideas for products that could not be discussed within the time frame of the interview (e.g., multi-player video games, virtual reality tools). Based on these observations, we consider the following input for the experiments: (1) the analysts should *document* the elicited requirements, and should be evaluated by the instructor based on the output produced, so that they will feel more compelled to perform their interviews in a more effective way; (2) a putative *budget* should be provided to customers, so that their idea will have some cost, and hence size, boundary; (3) the customers should provide ideas for mobile apps only, so that each interview concerns a specific, and simple, type of system.

L3. Time: according to the reviewers, part of the amount of time required for reviews was due to make sense of the type of ambiguity perceived, according to the classification provided [20]. This is witnessed also by the shorter time employed by reviewer 1, who was confident with the classification, with respect to reviewer 2, who was not. It was also observed that the classification was not sufficiently operational for the task. For these reasons, we consider more suitable to ask reviewers to report ambiguities based on a set of *question templates* (see Sect. V), which arguably represent typical questions that might be raised by the customer's words, and are defined by the authors based on an informal analysis of the linguistic patterns identifiable in the questions of our reviewers.

L4. Analyst's View: the ambiguities perceived by the analysts were identified by the two reviewers, since the analyst was not available. However, disagreement was observed on this task ($k_A, k_B < 0$). Furthermore, we do not know whether it is necessary to have external subjects to perform the review, or it is sufficient that the analysts themselves perform the task (RQ2). In the controlled experiment, we will ask also analysts to perform the review task to address these limitations.

V. RQ1, RQ2: Controlled Experiment

The goal of the controlled experiment is to perform a more rigorous evaluation of the updated review method, and to answer RQ1 and RQ2. To this end, we replicate the experiment at two universities, namely KSU and UTS. The experiment at KSU was conducted in February 2017, it involved 44 students, and we have to evaluate its results. The experiment at UTS will involve a comparable number of students, and will be conducted during March 2017. Compared to the exploratory study, the experiment differs on two main aspects: (a) the role of reviewers is played by students; (b) each interview is reviewed by the analysts who performed it, and by a reviewer.

A. Study Design

In this section, we outline the main items of the protocol defined to perform the controlled experiment. The complete protocol is available at https://goo.gl/PI2LLy. To address the limitation **L1** of Sect. IV-C, student customers, instead of researchers, will play the role of reviewers for interviews to which they did not participate.

a) Interview: The interview process will follow the same guidelines of the exploratory study (Sect. IV), with some integration to address part of **L2**. In particular, customers are

told: "Take a week to think about a mobile app for smartphones you would like to have developed. You have a $ 30,000 budget and your idea should be feasible within your budget. If the ideas you have seems not doable with this budget look at the apps you have on your phone and try to think how you would like to modify one of them."

b) Review: The review protocol is slightly different for reviewers and analysts, since these latter have to annotate their own interview, distinguishing between ambiguities perceived during the interview, and during the review. This helps to answer RQ2, and address **L4**. For the sake of space, here we report only part of the protocol for reviewers, which is similar to the one used in our exploratory study (Sect. IV-A). The only difference resides in step 3. Indeed, reviewers are not required to classify the ambiguities (**L3**), but are suggested the following guidelines to identify ambiguous fragments: "As a rule of thumb, stop the reproduction in any case in which, if you were the analyst, you would have asked the customer one or more questions of the form:"

- *What does it mean [...]?* (You have not understood the meaning of what you heard)
- *What is the purpose of [...]?* (You have not understood the purpose of what you heard)
- *Can you discuss in more detail about [...]?* (What you heard is too general)
- *You mentioned that [...], but [...]?* (What you heard contradicts what you heard before, or your vision of the problem)
- *Do you mean <A> or <B>?* (What you heard can mean different things)
- *I thought that with [...] you meant [...], was I wrong?* (You have doubts about a previous understanding of some concept)

c) Interview Output: To address **L2**, analysts are also required to provide requirements for their interview in the form of user stories (*As a <type of user>, I want <some goal> so that <some reason>*), and to write a list of human and time resources. The output is evaluated by the instructor.

B. Planned Evaluation and Threats to Validity

Our plan is to perform a quantitative evaluation of the experiment's results. In particular, we will evaluate the average degree of agreement between ambiguities perceived by analysts during the interview, ambiguities annotated by analysts during the review, and ambiguities annotated by reviewers. As for the exploratory study, also the average time for the review will be evaluated. Since the participants will also be asked their degree of competence in interviews, and in the domain of the interviews, these information will also be used to understand whether there is a correlation between experience, domain knowledge, and ambiguities perceived. To this end, typical regression analyses will be performed.

The population validity is the major threat that we foresee in this study, since we use students instead of practitioners to perform our interviews. Although according to Höst *et al.* [42] students with a good knowledge of computer science appear to perform as well as professionals, there is always a difference between the industrial world, and a role-playing settings. This limit will be addressed by the case studies, while other threats to validity, which we do not list here for the sake of space, have been considered in the experiment design.

VI. RQ3: Real-world Case Study

To answer to RQ3, about the usefulness of the elicitation questions defined through our method, we plan to perform a case study in industry, in which the method will be applied, and the impact of the questions will be monitored along the development. The idea is to gather qualitative data about the *perceived* usefulness of the questions produced after the first interview, and their *actual* usefulness observable after the delivery of the products. To identify our target companies, we are currently involved in an intensive dialogue with professionals from a small start-up company, and from a large company, who gave us feedback on the practicability of the idea. The start-up company suspects that the method might be too expensive in terms of time, since they normally employ a lightweight requirements process, and have a stronger focus on coding and prototyping. The larger company, instead, has a heavyweight requirements engineering phase, and see the potential benefits of the method. In addition, it was suggested to employ reviewers that belong to the firm, instead of researchers, so that potential confidentiality problems with the customers will be limited. For a larger company it seems also more realistic to allocate resources specifically on interview review. More focused guidelines for the case studies will be defined after the completion of the controlled experiments.

VII. Uncertainties, Unknowns and Risks

The proposed method entails a series of risks for its industrial applicability, which we briefly summarise below.

Domain Knowledge: the domain knowledge of analysts and reviewers may have an impact on requirements elicitation and review activities [41], [43], and also on the perception of ambiguity [21], [22], [41]. This suggests that subjects with specific knowledge (or *ignorance* [41]) might be more suitable than others for the role of reviewers. This aspect will be partially evaluated in our controlled experiment, while further research will enlighten which are the most adequate profiles for performing reviews of interviews.

Visual Aspects: we propose to review only the audio, and not the video as, e.g., Karras *et al.* [38], since customers may be uncomfortable in front cameras, while audio recording is less invasive. However, this implies that certain ambiguities that are resolved through other pragmatic means during the conversation, uselessly emerge during the review. In addition, during interviews, it is also common to draw diagrams, to facilitate communication. In future work, we foresee to integrate diagrams in the review, as well as video aspects. At the moment, we prefer to treat audio in isolation.

Cost: the proposed method is proportionally more expensive than the interview itself, in terms of time (Sect. IV). On the other hand, we argue that the duration of interviews is rather limited, compared to requirements analysis and documentation. Therefore, we foresee that the *absolute* time cost will

be acceptable in practice, if the benefits of the methods are tangible. Considering the cost in terms of number of persons in a team to be involved, we foresee that this aspect can be compensated by the higher level of internal knowledge sharing achieved through the review, as it happens for code review [5].

Incremental Development: interviews are performed at the beginning of a project, but additional interviews are normally carried out throughout the whole system life-cycle. The method will be validated on first interviews, while its effectiveness on later ones will be considered in future works.

VIII. Conclusion

Review of software artifacts, including requirements, is a common practice in software development. Nevertheless, requirements review may fail in its objective of improving the quality of the specification documents, since reviews are often based on flawed upstream requirements [8]. This paper proposes to bring review upstream in the software process to gather correct and complete information since the first elicitation interviews. To this end, we define a review method for interviews, in which ambiguities are detected to identify implicit information. Then, we outline our agenda to refine and assess the method by means of empirical studies.

References

[1] "IEEE Std 1028-2008 - IEEE Standard for Software Reviews and Audits," IEEE CS, 2008.

[2] O. Laitenberger and J.-M. DeBaud, "An encompassing life cycle centric survey of software inspection," *JSS*, vol. 50, no. 1, pp. 5–31, 2000.

[3] F. Shull, I. Rus, and V. Basili, "How perspective-based reading can improve requirements inspections," *Computer*, vol. 33, no. 7, pp. 73–79, 2000.

[4] A. Bacchelli and C. Bird, "Expectations, outcomes, and challenges of modern code review," in *ICSE'13*. IEEE, 2013, pp. 712–721.

[5] P. C. Rigby and C. Bird, "Convergent contemporary software peer review practices," in *FSE'13*. ACM, 2013, pp. 202–212.

[6] M. E. Fagan, "Design and code inspections to reduce errors in program development," vol. 15, no. 3, pp. 182–211, 1976.

[7] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, "Are the perspectives really different?: Further experimentation on scenario-based reading of requirements," in *Experimentation in Software Engineering*. Springer, 2012, pp. 175–200.

[8] F. Salger, "Requirements reviews revisited: Residual challenges and open research questions," in *RE'13*. IEEE, 2013, pp. 250–255.

[9] H. Femmer, B. Hauptmann, S. Eder, and D. Moser, "Quality assurance of requirements artifacts in practice: A case study and a process proposal," in *PROFES 2016*. Springer, 2016, pp. 506–516.

[10] F. Salger, G. Engels, and A. Hofmann, "Inspection effectiveness for different quality attributes of software requirement specifications: An industrial case study," in *Software Quality, 2009. WOSQ'09. ICSE Workshop on*. IEEE, 2009, pp. 15–21.

[11] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. Moreno, "Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review," in *RE'06*. IEEE, 2006, pp. 179–188.

[12] I. Hadar, P. Soffer, and K. Kenzi, "The role of domain knowledge in requirements elicitation via interviews: an exploratory study," *REJ*, vol. 19, no. 2, pp. 143–159, 2014.

[13] J. Coughlan and R. D. Macredie, "Effective communication in requirements elicitation: a comparison of methodologies," *REJ*, vol. 7, no. 2, pp. 47–60, 2002.

[14] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and managing software requirements*. Springer, 2005, pp. 19–46.

[15] V. Gervasi, R. Gacitua, M. Rouncefield, P. Sawyer, L. Kof, L. Ma, P. Piwek, A. De Roeck, A. Willis, H. Yang *et al.*, "Unpacking tacit knowledge for requirements engineering," in *Managing requirements knowledge*. Springer, 2013, pp. 23–47.

[16] A. Sutcliffe and P. Sawyer, "Requirements elicitation: towards the unknown unknowns," in *RE'13*. IEEE, 2013, pp. 92–104.

[17] A. Ferrari, P. Spoletini, and S. Gnesi, "Ambiguity cues in requirements elicitation interviews," in *RE'16*. IEEE, 2016, pp. 56–65.

[18] G. Rugg, P. McGeorge, and N. Maiden, "Method fragments," *Expert Systems*, vol. 17, no. 5, pp. 248–257, 2000.

[19] W. R. Friedrich and J. A. Van Der Poll, "Towards a methodology to elicit tacit domain knowledge from users," *IJIKM*, vol. 2, no. 1, pp. 179–193, 2007.

[20] A. Ferrari, P. Spoletini, and S. Gnesi, "Ambiguity as a resource to disclose tacit knowledge," in *RE'15*. IEEE, 2015, pp. 26–35.

[21] B. Rosadini, A. Ferrari, G. Gori, A. Fantechi, S. Gnesi, I. Trotta, and S. Bacherini, "Using NLP to detect requirements defects: an industrial experience in the railway domain," in *REFSQ'17*, ser. LNCS. Springer Berlin Heidelberg, 2017, vol. 10153, pp. 344–360.

[22] A. K. Massey, R. L. Rutledge, A. I. Anton, and P. P. Swire, "Identifying and classifying ambiguity for regulatory requirements," in *RE'14*. IEEE, 2014, pp. 83–92.

[23] L. Kof, "From requirements documents to system models: A tool for interactive semi-automatic translation," in *RE'10*, 2010.

[24] V. Ambriola and V. Gervasi, "On the systematic analysis of natural language requirements with Circe," *ASE*, vol. 13, no. 1, 2006.

[25] L. Mich, "NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA," *NLE*, vol. 2, no. 2, pp. 161–187, 1996.

[26] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in *RE'09*. IEEE, 2009, pp. 317–322.

[27] K. Pohl and C. Rupp, *Requirements engineering fundamentals*. Rocky Nook, Inc., 2011.

[28] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated checking of conformance to requirements templates using natural language processing," *TSE*, vol. 41, no. 10, pp. 944–968, 2015.

[29] D. M. Berry, E. Kamsties, and M. M. Krieger, "From contract drafting to software specification: Linguistic sources of ambiguity," 2003.

[30] S. Gnesi, G. Lami, and G. Trentanni, "An automatic tool for the analysis of natural language requirements," *IJCSSE*, vol. 20, no. 1, 2005.

[31] S. Tjong and D. Berry, "The design of SREE a prototype potential ambiguity finder for requirements specifications and lessons learned," in *REFSQ'13*, ser. LNCS. Springer, 2013, vol. 7830, pp. 80–95.

[32] B. Gleich, O. Creighton, and L. Kof, "Ambiguity detection: Towards a tool explaining ambiguity sources," in *REFSQ'10*, ser. LNCS, vol. 6182. Springer, 2010, pp. 218–232.

[33] H. Femmer, D. M. Fernández, S. Wagner, and S. Eder, "Rapid quality assurance with requirements smells," *JSS*, vol. 123, pp. 190–213, 2017.

[34] F. Chantree, B. Nuseibeh, A. N. D. Roeck, and A. Willis, "Identifying nocuous ambiguities in natural language requirements," in *RE'06*, 2006, pp. 56–65.

[35] H. Yang, A. N. D. Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Analysing anaphoric ambiguity in natural language requirements," *Requir. Eng.*, vol. 16, no. 3, pp. 163–189, 2011.

[36] A. Aurum, H. Petersson, and C. Wohlin, "State-of-the-art: software inspections after 25 years," *Software Testing, Verification and Reliability*, vol. 12, no. 3, pp. 133–154, 2002.

[37] A. Katasonov and M. Sakkinen, "Requirements quality control: a unifying framework," *REJ*, vol. 11, no. 1, pp. 42–57, 2006.

[38] O. Karras, S. Kiesling, and K. Schneider, "Supporting requirements elicitation by tool-supported video analysis," in *RE'16*. IEEE, 2016, pp. 146–155.

[39] H. Sharp, Y. Rogers, and J. Preece, *Interaction Design: Beyond Human Computer Interaction, 4th edition*. John Wiley & Sons, 2015.

[40] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[41] A. Niknafs and D. M. Berry, "An industrial case study of the impact of domain ignorance on the effectiveness of requirements idea generation during requirements elicitation," in *RE'13*. IEEE, 2013, pp. 279–283.

[42] M. Höst, B. Regnell, and C. Wohlin, "Using students as subjects, a comparative study of students and professionals in lead-time impact assessment," *ESE*, vol. 5, no. 3, pp. 201–214, 2000.

[43] Ö. Albayrak and J. C. Carver, "Investigation of individual factors impacting the effectiveness of requirements inspections: a replicated experiment," *ESE*, vol. 19, no. 1, pp. 241–266, 2014.