



**HAL**  
open science

# Examining Requirements Documentation through the Focused Conversation Method

Hamna Aslam, Alexandr Naumchev, Jean-Michel Bruel, Joseph Brown

► **To cite this version:**

Hamna Aslam, Alexandr Naumchev, Jean-Michel Bruel, Joseph Brown. Examining Requirements Documentation through the Focused Conversation Method. 29th International Conference on Information Systems Development (ISD 2021), Sep 2021, València, Spain. hal-03435282

**HAL Id: hal-03435282**

**<https://hal.science/hal-03435282>**

Submitted on 22 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Examining Requirements Documentation through the Focused Conversation Method

***Hamna Aslam***

*Université de Toulouse IRIT-CNRS  
and Innopolis University  
Innopolis, Russia*

*h.aslam@innopolis.ru*

***Alexandr Naumchev***

*Innopolis University  
Innopolis, Russia*

*a.naumchev@innopolis.ru*

***Jean-Michel Bruel***

*Université de Toulouse IRIT-CNRS  
Toulouse, France*

*bruel@irit.fr*

***Joseph Alexander Brown***

*Innopolis University  
Innopolis, Russia*

*j.brown@innopolis.ru*

## Abstract

Requirements documentation comprehends the challenge of accurately interpreting project details as specified by the client. There is no standard template that ensures that documentation would be evident in all aspects to developers. Furthermore, requirements documentation continuously evolves and has a continuously evolving audience (changes in the development team, reusing requirements in a new project, changing sub-contractor, etc.).

In contrast, some well-known techniques, such as the Focused Conversation Method, have been proven to help determine needed direction and decisions, especially when it involves people of mixed backgrounds. We hence had the intuition to use such an approach to the requirements documentation phase.

Based on the focused conversation method, consisting of a mix of interviews and surveys analysis, we argue that the outcomes (organized and structured requirements) are efficient in terms of sound qualities (i.e., unambiguous).

The approach is evaluated on an empirical case study with students of the Innopolis University.

**Keywords:** Focused Conversation Method, Requirements Engineering, Requirements Elicitation

## 1. Introduction

A successful product delivery relies on the constructive requirements engineering phase. The interview with the stakeholders and documentation defines the project for the development team. During the development phase, concerns arise as certain ambiguities in project requirements only become visible once the development starts. Specific approaches have been introduced for requirements elicitation to achieve a thorough understanding of the system among all stakeholders. Some of the techniques include introspection, task analysis, card sorting, and laddering, etc. [12]. Moreover, Denger and Olsson [5] survey existing requirements quality assurance (QA) techniques. They distinguish constructive versus analytic requirements QA techniques. Constructive techniques produce new requirement artifacts. Analytic techniques work with already existing artifacts, with no creation of new ones. Denger and Olsson then break these

two broad categories into more complicated ones. None of these categories mentions focused conversation as a requirements QA technique.

The quality of interaction among project stakeholders is crucial to ensure productive results. However, another perspective regarding achieving complete and exceedingly understandable documentation is to reflect upon the documentation systematically. A systematic examination can be conducted by reading the documentation and pointing out inconsistencies and ambiguities. This overview has the potential to miss vital points. Therefore, we refer to a questioning methodology known as *focused conversation method*, which enables detailed analysis of the object or process under examination.

The focused conversation method was first developed by United States Army Chaplain, and Art Professor Joseph Mathews as a method of art appreciation and reflection [8]. This method includes four categories of questions, *Objective, Reflective, Interpretive, and Decisional*. Hence, it is sometimes also referred to as the ORID set of questions or framework. The objective questions identify facts, such as the participants clearly understanding what happened in the process. Examples of objective questions include what you noticed, what your task was, what was done. The next category of questions is reflective questions. The reflective questions identify emotions associated with what happened during the process (i.e. the situation being examined). Examples of reflective questions include, what caused the confusion, where you felt anxious, what was understandable and easy, what was difficult. Further, the interpretive questions identify the implications and the significance of informed events in context to the objective and reflective questions. Some examples include what do you recommend, what is the significance of each step in the process, what are your insights, etc. The last phase or set of questions is decisional. The decisional questions state the conclusions, further actions, and further plans. Examples of decisional questions include how you see the process, what you will change, what you learned, the next step, etc. Overall, the method enables us to examine any process systematically through guided conversations with the stakeholders involved. Further, each element of the process and corresponding affordances are identified.

The focused conversation method is one of the highlighted elements for Instructional Skills Workshops [4] [7] [11] where it is used in order to engage students in reflective practices. While it has been primarily used in the humanities and in its original application as a method of art appreciation, uses have been found in design problems as a reflective practice, see for example [2], where the method was used to examine the creation of rules for a game. Furthermore, Calabrese [3] refers to a focused conversation method as a useful tool to achieve structured conversations among teams in an agile framework. Peisan [10] considers the incorporation of focused conversation methodology to significantly enhance the quality of in-depth interviews in the field of consumer research.

*Our research work applies focused conversation methodology to examine the affordances of the requirements document for the developers. Therefore, to identify aspects of the requirements document, leading to clarity, confusion, ambiguity, etc.*

The research has been conducted with students of requirements engineering course. As the research is conducted in a classroom setting, all students were involved in the same process, and hence there is no control group. The students worked as teams to develop projects as part of the course assignment. The focus of the research is on examining the requirements documentation. We have simulated a frequently occurring industry scenario in which a project is handled by various people as the development process goes on. This poses additional challenges for the new developers as they have to understand the system majorly through the requirements document. Therefore, in order to simulate this, the projects were shuffled among teams at different stages of development. The research involves *focused conversation method* based interview with the teams to examine the clarity of requirements documents.

The remainder of the paper proceeds as follows. Section 2 discusses the study design as well

as describes the course and projects undertaken by students. Section 3 details the questions for the focused conversation method and provides a summary of the responses. Section 4 provides discussion and Section 5 draws conclusions of the research.

---

## **2. Study Design**

The case study has been conducted with the students of the “Requirements Engineering” (RE) course of the one-year “Master of Science in Information Technology - Software Engineering” (MSIT-SE) program at Innopolis University. The course involved 26 students with several years of industrial experience. Some students have managerial experience. Students had 12 hours per week for course-related activities. The course syllabus largely relies on two textbooks:

- “Requirements Engineering – From System Goals to UML Models to Software” by Axel van Lamsweerde [9].
- “Advanced Use Case Modeling: Software Systems” by Frank Armour and Granville Miller [1].

### **2.1. Prerequisites to the Course**

As a starting course of the MSIT-SE program, the RE course required the same prerequisites as the MSIT-SE program in general. The MSIT-SE degree is designed for junior professionals with one to two years of work experience in software development who want to boost their careers and become technical leaders, software architects, or project managers.

### **2.2. Objectives of the Course**

The course had the following key objectives:

- To introduce the motivation, conceptual background, and terminology on which requirements engineering relies.
- To provide a comprehensive account of state-of-the-art techniques for requirements engineering.
- To let the students experience the actual requirements-caused problems faced by real software teams.

### **2.3. Student Projects**

The 26 students were divided into 8 teams – 6 teams of 3, plus 2 teams of 4. The 8 teams were assigned 4 project proposals coming from the teams themselves. The students chose the 4 proposals out of the 8 presented through anonymous voting. Each of the 8 teams then was assigned one of the 4 proposals till the end of the course. Every 2 teams responsible for the same proposal were free to interpret the proposal differently during the project’s execution. Each team had to create a GitHub repository to store all the artifacts related to the assigned proposal. Each team was in charge of traceability, completeness, and consistency of the artifacts inside their repositories.

### **2.4. The Proposals**

The students chose the following 4 proposals to work on:

**FoodDelivery** – a service to let Innopolis University students have stable food delivery on time with planned meals on a daily basis.

**RoomServiceOnDemand** – a service that lets people living in the dormitories order and schedule the room cleaning service.

**BBQReserver** – a service letting Innopolis city residents create and manage bookings of the only barbeque area.

**InnoCalendar** – a service was extracting elective course information from Google Docs to Google Calendar for individual course selection.

## 2.5. Execution of the Projects

The key idea behind the chosen scheme of the project execution (Figure 1) was to expose the students to the most annoying problems of industrial projects. To achieve this goal, we decided to mix the customer-developer pairs between consequent iterations of the projects. Also, each team was simultaneously playing two roles: they were both developers and customers to some other team. We had allocated 8 project teams, and each of them then had to describe and present an idea for a project. After having all the proposals presented, the students chose 4 best proposals by means of voting.

The project consisted of the following three iterations:

1. Analysis, consisting of requirements documentation in the form of use cases [6].
2. Implementation, consisting of implementing the elicited requirements in the form of a minimum viable product (MVP) with only the base cases implemented.
3. Testing, consisting of extending the MVP to cover the remaining use cases and verifying the resulting implementation.

For each of the 3 iterations, the 8 teams were split into 4 pairs of 2 teams. Inside each pair  $(T_1, T_2)$ ,  $T_1$  would work on the proposal assigned to  $T_2$ , and  $T_2$  would work on the proposal assigned to  $T_1$ .  $T_1$  would put the resulting artifacts to the repository owned by  $T_2$ , and vice versa. At the end of each iteration, the teams had to present their results. After the presentations, the pairwise distribution of the teams was changed for the next iteration. The idea was to let the teams understand the problems associated with changing the contractor on the one hand, and the customer – on the other hand. For the second iteration, assuming  $T_1$  is now assigned new partner team  $T_3$ ,  $T_1$  had to implement an MVP for the  $T_3$ 's proposal using the requirements elicited for  $T_3$  by their partner team during the first iteration.  $T_3$  had to do the same for  $T_1$ . A similar scheme applies to the testing iteration. The project flow is shown in Figure 1.

We conducted the focused conversation method-based survey at the very beginning of the second iteration. Inside each pair  $(T_1, T_3)$ ,  $T_1$  had to reflect on the requirements document elicited for  $T_3$  by their partner team during the first iteration, and vice versa.

During the second iteration, each team was implementing MVP for their new customer team. The requirements for the MVP were elicited by someone else, which is a realistic situation. We found it valuable for the teams to reflect on the requirements they have never seen before. This activity, when performed before jumping into coding, may save a tremendous amount of time.

## 2.6. Focused Conversation Method

The focused conversation method can be adopted in any context where contemplation is required. The method is led by a facilitator that guides the discussion forward and ensures that each person in the group is on the same page about interpreting the questions being asked. It replicates the natural thought process in which we recall an event, associate feelings, develop interpretations, and make decisions. Starting with objective questions, any scenario under investigation is analyzed upon. The objective questions are designed to obtain a recap of the scenario

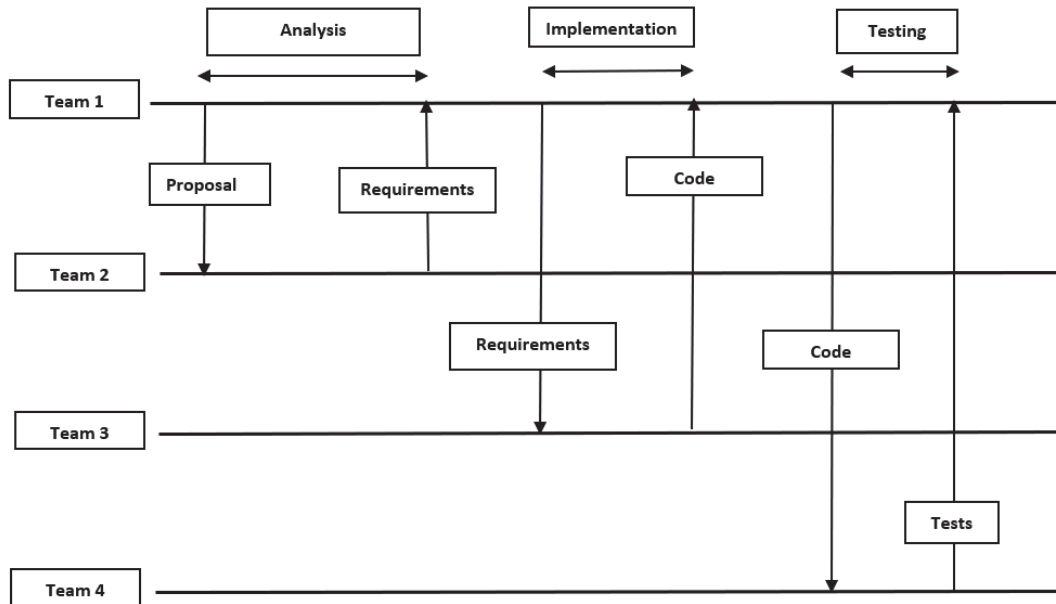


Fig. 1. Project execution among different Teams

being discussed. Objective questions are meant to collect information about the scenario, such as what happened or what do we know about the issue at hand.

The discussion proceeds to reflective questions. Reflective questions such as, how does this make you feel; identify feelings associated with what happened. These feelings can be subjective and give us a sense of what people are concerned about. The interpretive questions that follow are essential in determining the situation's essence after understanding the objective facts and subjective feelings associated with these facts. Interpretive questions put the discussion in perspective. Examples of interpretive questions include what it means to you, and what the information shows, etc. The last set of questions, called decisional questions, lead to deciding the future course of action based upon what information is attained from earlier questions. The questions can be, what decisions we can make and what are the next steps etc.

The focused conversation or ORID method allows the systematic reflection on a situation. Consequently, it inhibits occurrences where hasty decisions are made without careful thought and consideration of the best options available. This method is applicable in a variety of contexts. The requirements document is also an application area for the ORID method, as development teams have to be on the same page about the decisions specified in the document. Any concern raised by the developers related to the requirements document is addressed as we proceed from the objective to the reflective, interpretive and decisional questions. We understand the problem, how it affects us, the causes, and how they may be addressed through this systematic approach.

### 3. Examining Requirements Document: Focused Conversation Based Questionnaire

The questionnaire is adopted from the template titled, *Evaluating the Progress of a Project* [8], and customized according to course needs. The questions were given during the lecture. The students were asked to go through the requirements document before the lecture. The questions were required to answer as a team, and the time given to answer each question was four minutes.

As the method includes Objective, Reflective, Interpretive, and Decisional questions [ORID],

they are stated as follows:

### ***Objective Questions***

This set of questions examines the subjective perceptions of the real scenario of document preparation. This section of the questioning aims to have a shared idea of what the document states.

1. As you read the requirements document, what statements or points caught your attention?
2. Where was the document clear?
3. Where was the document not clear?
4. What other information do you need?

### ***Reflective Questions***

The reflective questions examine associated emotions to clarify experiential goals.

1. What surprises did you find in the documents?
2. What parts of the document were easy to read?
3. What parts of the document were difficult to read?
4. Where have you gotten stuck?

### ***Interpretive Questions***

The interpretive questions examine the meaning and implications of the outcomes.

1. What appears to be the key issues or key problem areas?
2. Where will you need extra help?
3. What kind of help will you need?
4. What questions do you need to get answered?

### ***Decisional Questions***

In the end, a decision about the document needs to be made, and new tasks need to be set.

1. What changes are you recommending to the document?
2. What changes are you recommending to the working process?
3. What are the next steps?
4. Who needs to do these steps?

## **3.1. Summary and Analysis of Data**

The questions are answered as a team. Each team has studied the requirements document for another team. The answer to the questions are detailed as follows:

### *Objective Questions*

**1. As you read the requirements document, what statements or points caught your attention?**

– Seven teams mentioned that use case diagrams caught their attention. They were either missing some actors or having inconsistency in alternative cases. One of the teams (analyzing “InnoCalendar” project requirements) stated [sic]:

*“In the Use Case “Add new lectures” secondary actor is missing. In the Use Case “Update schedule,” the 3rd non-behavioral requirement could not be met, as there is no possibility to be always in sync with other system with the current schedule.”*

**2. Where were the documents clear?**

– In terms of clarity, all teams referred to the broader details of the system.

**3. Where are the documents not clear?**

– The teams referred to the lack of glossary. The lack of glossary led to the difficulty of understanding use case terms. Furthermore, they also pointed out the inconsistency of use cases of alternative flow.

**4. What other information do you need?**

The teams referred to the following aspects regarding the information they require: glossary, use case diagrams with precise actor arrangements, and missing requirements.

### *Reflective Questions*

**1. What surprises did you find in the documents?**

– The teams referred to the lack of terms in the glossary. Furthermore, they referred to the requirements from the clients that have unreasonable specifications, such as limiting the number of notification recipients. They also informed about the colors used in the use case diagram. There is no indication, for example, as to whether colors have a specific meaning.

**2. What parts of the document were easy to read?**

– The phrase *easy to read* was perceived as the document’s simplicity in terms of readability. Participants referred to the overall document, including diagrams, as an easy read in general.

**3. What parts of the document were difficult to read?**

– The transition of events in the use case diagram introduced difficulty as there were some missing logical connections between event flows.

**4. Where have you gotten stuck?**

– Teams referred to similar points as explained above. The incomplete glossary and ambiguity in use cases needed clarification.



### *Interpretive Questions*

**1. What appears to be the key issues or key problem areas?**

– The key problem areas include the absence of exceptions, incomplete specification of alternative flows, lack of clarity upon owners of administrative tasks, and wrong implementation of use cases.

**2. Where will you need extra help?**

– Teams mentioned the need to clarify technical terms that are not stated in the glossary, some details regarding the interface design, and clarity upon ambiguous requirements.

**3. What kind of help will you need?**

– Teams stated the need to have an additional interview with the customers to clarify project details.

**4. What questions do you need to get answered?**

– The questions informed by the teams are related to the project requirements. One of the teams did not have any questions for the stakeholders. Some of the questions mentioned by other teams are stated as follows [sic]:

- *When to send a notification?*
- *What should be included into the notification?*
- *Is “University Education Department” really an actor in the system?*
- *What if the bot fails in notifying the user?*

### *Decisional Questions*

**1. What changes are you recommending to the document?**

– The changes recommended by teams were about issues they encountered in understanding the system via use cases and text. Such as, as stated by participants [sic]:

- *Remove Server from use cases*
- *Clarify Appendix diagrams*
- *Put Glossary into the table*
- *Direct arrows from actors to use cases*

**2. What changes are you recommending to the working process?**

– Three teams recommended no changes in the working process. The other five teams suggested the need for more meetings and to establish clarity upon the tasks of each team member. The teams also emphasized the need for allocating extra time in the process of requirements understanding. Therefore, to have enough time to clarify points with the customer not obvious from the document.

**3. What are the next steps?**

– The teams were majorly on the same page in regards to their next steps. One of the team stated [sic]:

- *One more meeting with the stakeholder before starting the development of process.*
- *Designing and prototyping.*

Another team specified their next steps as [sic]:

- Deal with problems identified in the requirements document via interaction with customer
- Evaluate deadlines and identify optional requirements
- Identify a scope for implementation
- Select an appropriate process for development
- Divide the work (create tasks)
- Make plans - interact with the customer
- Start to implement
- Interact with the customer during the implementation

#### 4. Who needs to do these steps?

- The teams referred to themselves for the further implementation of the project. The customer would be involved in the project review phase.

### 3.2. Discussion on ORID Questionnaire Responses

The questionnaire has been presented at the beginning of the implementation phase. Teams in the implementation role answered this questionnaire, and therefore the requirements document they got was written by another team. These teams were also not involved in the interview process for the project they got documentation on. The requirements document is the major resource each team had to begin the implementation with.

The answers to objective questions pointed out the inconsistencies and ambiguities in the documentation. Students referred to the missing elements of the use case diagrams, incomplete glossary, the inconsistent flow of events, etc. The reflective questions were answered based on objective questions, such as the missing elements of diagrams and ambiguous descriptions of functionalities posed difficulty in understanding the system flow.

The interpretive questions helped teams to understand how they can solve the problems they have encountered. The students stated the questions they need to clarify from the customers before starting the implementation. Therefore the interpretive questions suggest actions to be taken to improve the understandability and clarity of the requirements document. Furthermore, answering the decisional questions provided clarity upon what needs to be changed in the workflow and what are the future actions to improve the document.

The teams reported that answering the ORID questionnaire helped them understand the requirements document systematically. They could identify problem areas and determine the points that need attention before implementation. The ORID questionnaire was expressed to be extremely useful at this stage of the project. Based on the answers to the questionnaire, the teams organized a meeting with the project's customers to understand the project concretely.

This study focused on the requirements document. However, the ORID method is applicable to reflect on the whole process and not just the requirements document. The questions will be different depending on the phase or artifact being addressed.

## 4. Discussion

The students had to conduct elicitation sessions individually in the first iteration of the project. Making the students reflect on their documentation experience gave us a retrospective on the quality of the documentation process itself. Documentation is supposed to ensure the following qualities of requirements [5]:

- Comprehensibility, through developing a common terminology.
- Completeness, by identifying all the relevant stakeholders and their respective needs.
- Verifiability and feasibility, through involving the testers and the developers.

- Correctness, through making the documentation process driven by the business concerns.

It was up to the students to interview. Some students decided not to interview members of the then-current customer team but interview an external stakeholder. After having the interviews finished, the students then had to specify, in teams, the elicited information as to use cases and push the use cases to the GitHub repository of the customer team. The use cases would then be passed to the developers for the implementation phase. For the implementation phase, after reshuffling the pairwise allocation of the teams, they would implement the use cases found in the GitHub repositories of their new customers. We conducted ORID sessions among the teams to let them quickly start working with the newly assigned use case documents. The teams identified many issues in the analyzed documents during these sessions. The identified issues may serve as input for the creation of new requirement templates.

Templates and standards provide elements that should be specified when documenting the requirements. They contribute to better requirements in the following ways [5]:

- Completeness, through requiring the practitioner to adhere to the template or standard.
- Understandability and modifiability, through ensuring the uniform structure of the resulting documents.

ORID sessions, when applied to actual requirements documents, facilitate the development of quality templates by identifying aspects of the actual documents to which their consumers pay the most prominent attention. In the present research, we identify the important, from the actual developers' viewpoint, parts of a requirements document. Surveying and ORID ensure that elicitation and specification, when perceived as requirements QA techniques, bring a better understanding of the system-to-be.

In our study, we applied the ORID focused conversation method as an inspection-based analytical RQA technique [5]. The students have inspected an existing use case document, which they had never seen before, from the future developers' standpoint. We were providing the questions to them one by one during the study. In this way, they had a chance to focus on one question at a time when inspecting the document.

## 5. Conclusion

The research work presents an empirical study in which the focused conversation method has been practiced to evaluate requirements documents. The requirements documentation for different projects was shuffled among teams. Therefore, each document was read by the team of developers who were not involved in the interview and the document's writing. The goal is to investigate the understanding of the document to identify misleading aspects. The study outcomes suggest that the focused conversation method has proven helpful for the developers to determine the clear or ambiguous points about project requirements. The developers could concretely pinpoint the areas that needed their attention before the implementation starts. The focused conversation method also helped developers identify further steps to rectify problem areas and who should address these points.

The research work demonstrates that the focused conversation method can provide a thorough analysis of the requirements document. This process, if practiced before the implementation phase, can lead developers towards clarity on system requirements.

For future research, we consider applying the focused conversation method process in the industrial project development setting. We would also like to compare the focused conversation method with other tools for reflecting and validating software requirements; such a comparison exceeds the scope of the present work, which is an experience report.

## References

1. Armour, F. and Miller, G. (2001). *Advanced Use Case Modeling: Software Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
2. Brown, J. A. (2019). A focused conversational model for game design and play-tests. In *Games and Learning Alliance conference*.
3. Calabrese, J. (December 1, 2014). Facilitating with the focused conversation. <https://agileforall.com/facilitating-with-the-focused-conversation/>, Online; accessed 09 November 2019.
4. Dawson, D., Borin, P., Meadows, K., Britnell, J., Olsen, K., and McIntyre, G. (2014). The impact of the instructional skills workshop on faculty approaches to teaching. Technical report, Higher Education Quality Council of Ontario.
5. Denger, C. and Olsson, T. (2005). *Quality Assurance in Requirements Engineering*, pages 163–185. Springer Berlin Heidelberg, Berlin, Heidelberg.
6. Jacobson, I. (1993). Object-oriented software engineering - a use case driven approach. In *TOOLS (10)*, page 333. Prentice Hall.
7. Macpherson, A. (2012). *The instructional Skills Workshop as a transformative learning process*. PhD thesis, Simon Fraser University, Burnaby, BC.
8. Stanfield, R. B. (2000). *The Art of Focused Conversation: 100 Ways to Access Group Wisdom in the Workplace*. New Society Publishers.
9. van Lamsweerde, A. (2009). *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley.
10. Yu, P. (2005). A focused conversation model in consumer research: the incorporation of group facilitation paradigm in in-depth interviews. *ACR Asia-Pacific Advances*, 6, eds. Yong-Uon Ha and Youjae Yi, Duluth, MN : Association for Consumer Research:337–344.
11. Zhirosh, O., Brown, J. A., and Tickner, D. (2019). Democratizing faculty development - establishing a training program at a new computer science university in russia. In *ASEE Annual Conference and Exposition, Conference Proceedings*, Tampa, Florida.
12. Zowghi, D. and Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and managing software requirements*, pages 19–46. Springer.