sciendo RIGA TECHNICAL UNIVERSITY

Applied Computer Systems

ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2019, vol. 24, no. 2, pp. 141–149
https://doi.org/10.2478/acss-2019-0018
https://content.sciendo.com

Results From Expert Survey on System Analysis Process Activities

Laima Leimane[1], Oksana Nikiforova[2*]
[1, 2] *Riga Technical University, Riga, Latvia*

***Abstract* – System analysis is a crucial and complex step in software engineering process, which affects the overall success of the project and quality of the project outcome. Even though Agile methods have become widely popular, these methods have no structure when it comes to requirements elicitation and specification, which can have impact on whether a project has favourable outcome. Nevertheless, regardless of the chosen approach by industry practitioners, it is important to identify, which activities are currently performed, and analyse the causes and possible issues, which are encountered. The paper presents results from expert survey on the importance of activities related to requirements elicitation, analysis and specification process and the use of tools to support this process. Delphi method, which is used to evaluate the responses, is described. Lists of activities are ranked according to importance and additional information on expert responses is given in the paper. The information can give an insight into the activities and tools that are used in the industry.**

***Keywords* – Delphi method, expert survey, framework, system analysis.**

I. Introduction

Regardless of methodology or approach used in software development projects, to a lesser or greater extent the system analysis is performed. Traditional requirements engineering approach consists of requirements elicitation, analysis, documentation and management, and in this approach all requirements are gathered and analysed at an early stage [1]. Gathered requirements are described in highly detailed documents and are non-negotiable [2]. One can conclude that if all requirements are gathered exclusively at the beginning stages of the project, changes suggested later could potentially be difficult to implement. In contrast, Agile methodologies embrace changes and value working software over comprehensive documentation [3], which could benefit projects with dynamic, changing business domain and processes. This flexible approach has gained notable popularity since it was introduced in the 2000s [4]. When it comes to documenting requirements, it is reported that the most popular requirements notation is a user story, which generally has the following structure: "As a <type of user>, I want <goal>, [so that<reason/benefit>]" [5]. This notation helps identify who is going to use the function to be developed, what the desired outcome is and why it is needed. As stated before, Agile approach embraces changes and elicits requirements only when needed – just before development [6]. If compared by the manner, requirements engineering is performed using a traditional approach; it is limited in time, finite process is performed only at initial stages of project, while in Agile – it is a continuous, extensive process that is performed throughout the entire software development process [6].

Studies show that quality and precision of requirements directly impact the success of the overall project [7]. Researchers have reported examples of failed projects due to neglect of non-functional requirements and some of the failed projects have devastating financial consequences, for example, a $2.7 billion tool for U. S. Army intelligence was discarded as it had significant issues related to performance, usability and capacity [8], [9]. Another study in the UK has reported that 40 % of overall issues of software development process are related to requirements engineering process [10]. Neglect of non-functional requirements is a significant issue in Agile projects as if neglected, 60 % of projects fail [8]. Researchers suggest that almost 50 % of user stories created by practitioners in industry have quality defects [11] as most requirements formulated in natural language are lexically, syntactically, semantically or structurally ambiguous [12]. It is evident that there are many issues in this process and consequences can be both minor and tremendous.

In order to eliminate consequences of known issues, certain actions should be performed by the practitioners. To identify the possible activities in requirements elicitation, analysis, specification and validation process, a mapping table was created [13] and used as a basis for a survey conducted among industry professionals. The results from the survey were evaluated using Delphi method [14] and provided an insight into the activities that were most frequently used and the tools that supported the system analysis [15].

The paper is structured as follows. Section II provides an overview of the conducted survey and Delphi method, which has been used for result evaluation. Section III presents the results of expert survey. Section IV provides an overview of the related work. Finally, Section V contains the authors' conclusions and suggests possible areas for the future research.

II. Expert Survey on System Analysis Activities

As mentioned before, the created mapping table [13] was taken as a basis for the expert survey. Similar activities were merged and some were excluded from list [15]. One example would be such activities as Requirements documentation, creating Software Requirement Specification, User stories,

*Corresponding author's e-mail: oksana.nikiforova@rtu.lv

which were combined under one activity – Requirements documentation with options to choose from. Participants of survey could choose as many options as they deemed to be appropriate and could also additionally write their own answer if it was not provided by the survey [15].

The chosen topic is very specific and in order to evaluate the frequency of every activity the expert should have a certain level of expertise in working with requirements, so the survey was distributed among relevant experts of similar competence [15]. For evaluating and forming the expert group, Delphi method was chosen [15]. Delphi method is used to collect knowledge, experiences of experts and is especially useful when the studied phenomenon is rather complex or the topic is complicated to define or the number of experts is relatively small [16]. Number of participants is not confined [16], but is recommended to consist of 15 to 20 experts [14] as the main objective is to gather data from relevant experts with appropriate competence [16], [17]. Delphi method usually consists of several iterations as it can be used to reach consensus among experts [16], but in this case only one iteration was conducted as the objective was to determine the current situation [15].

Experts were given a survey, which was created in Google Forms. Survey was anonymous, which was essential for Delphi method [16]. Survey consisted of 34 questions of which 33 were activities and one additional question was about the use of tools [15]. Every activity was to be evaluated using the rating scale.

TABLE I
RATING SCALE FOR ACTIVITIES

Rating	Description
0	Not performed in any project
1	Performed rarely
2–3	Done sometimes
4–5	Activity can be done but it is not so important
6–7	A fairly important activity that may not be performed in some circumstances
8–9	Important activity that is rarely omitted
10	Performed in each project. Unable to skip this activity

Survey was distributed among relevant experts, who were asked to consider giving this survey to their colleagues with similar competence. Each expert was asked to confirm that they were participating in requirements elicitation, analysis and specification process [15]. In case an expert's response was negative, their respective survey results would be removed from the analysis. The design of survey was created so that several questions would overlap [15]. The intent was to evaluate responses and determine if they were valid for further analysis. Example would be three activities from the survey [15] – requirements elicitation, elicitation of functional requirements and elicitation of non-functional requirements. These activities overlap and requirements elicitation is a more general activity. One expert whose response was excluded from further analysis rated the general activity with 5, elicitation of functional requirements with 7 and elicitation of non-functional requirements with 6 [15], which could be analysed using a different kind of evaluation method, possibly determining attitude towards or perception of activities. Responses, which contained blank answers, were also excluded as they were not valid for Delphi method.

Overall 49 experts participated; 17 of them were chosen with the most consistent views on this topic. Due to a large number of questions, they were divided into three groups [15]:

- The first group – questions related to requirements elicitation and analysis;
- The second group – questions related to requirements specification;
- The third group – questions related to requirements validation and other activities (use of tools, demo creation, creation of visual materials).

Value of importance G_j of activity j was calculated using [15]:

$$G_j = \sum_{i=1}^{m} R_{ij}. \quad (1)$$

Mean value of importance was calculated as follows:

$$\bar{G} = \frac{1}{n}\sum_{j=1}^{n} G_j. \quad (2)$$

Deviation of activity j from mean d_j was calculated as follows:

$$d_j = \left|G_j - \bar{G}\right|. \quad (3)$$

Sum of square deviations was calculated as follows:

$$G = \sum_{j=1}^{n} d_j^2. \quad (4)$$

Number of equal ratings t_i by expert was used to calculate parameter T_i as follows:

$$T_i = (t_i^3 - t_i). \quad (5)$$

For every question, the group coefficient of coherence K was calculated:

$$K = \frac{12G}{m^2(n^3-n) - m\sum_{i=1}^{m} T_i}. \quad (6)$$

Expected range for coefficient K is between 0 and 1, meaning the closer K is to 1, the higher the level of coherence among experts [15]. If the value of coefficient is close to zero, additional experiments should be conducted as the level of coherence among experts has not been achieved [14].

III. RESULT ANALYSIS

As previously stated, the level of coherence was evaluated among 17 industry experts using equations described in the previous section. Table presents overall results for every question group [15]. It can be concluded that the level of coherence among experts is high enough to further analyse and make conclusions from the results and it is not necessary to

modify the existing survey and to conduct another iteration of this process [15].

TABLE II
OVERALL RESULTS OF THE LEVEL OF COHERENCE

Question group	Level of coherence *K*
The first group (elicitation, analysis)	0.59
The second group (specification)	0.58
Third group (validation, others)	0.62

One of the results of this survey is a list of activities sorted by their importance according to the experts. List for the first group [15] can be seen in Table III.

TABLE III
THE FIRST GROUP SORTED BY IMPORTANCE

	The first group (K = 0.59)	Importance of activity (G_j)
1.	1. Requirements elicitation	45.0
2.	3. Eliciting functional requirements	61.5
3.	7. Requirements analysis	67.0
4.	12. User role identification	86.0
5.	13. Stakeholder and/or user identification	101.0
6.	5. Conducting interviews	116.5
7.	9. Business process and function identification	130.5
8.	14. Determining how and which data are used in the process	133.5
9.	2. Using business process models for requirements elicitation	135.0
10.	8. Domain identification and analysis	142.5
11.	10. Business rule identification	171.5
12.	15. Identifying legal regulation risks and other known risk factors	172.5
13.	4. Eliciting non-functional requirements	201.5
14.	6. Survey	221.0

From the obtained results, one can conclude that experts deem requirements elicitation and eliciting functional requirements as important activities but eliciting non-functional requirements is not considered of the same importance [15]. It correlates with the current research, which highlights issues regarding neglect of non-functional requirements [8], [9], [18]. Another possibility is that elicitation of non-functional requirements is performed by other project team participants as elicitation of non-functional requirements requires expertise in topics such as security, usability, performance and portability [18]. If experts work with already established platforms, software, tools, which already have characteristics regarding usability, security, performance etc., then it would explain why these experts do not consider eliciting non-functional requirements as important. Requirements analysis, user role identification and stakeholder and/or user identification are considered to be important activities as well as conducting interviews. Less important activities, according to this expert group, are related to business processes, business rules, legal regulations and domain analysis, which could imply that often business processes and business rules could be undefined, not documented. Given how popular Agile methodologies have become, the use of surveys for requirements elicitation has been established as the least important activity because Agile approach suggests that conversation is the key tool that ensures communication of requirements [2]. This also seems probable considering that stakeholder identification and conducting interviews were rated as important according to the experts. Surveys mentioned in literature [13], [19] have limitations when it comes to observing behaviour of participants and discovering new information through conversations. Surveys can be helpful when users/stakeholders are in different time zones, as preserving anonymity is important and the number of survey participants is large.

List for the second group of activities can be seen in Table IV [15].

TABLE IV
THE SECOND GROUP SORTED BY IMPORTANCE

	The second group (K = 0.58)	Importance of activity (G_j)
1.	16. Requirements specification	46.0
2.	28. Model system structure	56.0
3.	21. Requirements documentation	57.5
4.	19. System boundary and scope analysis	84.0
5.	29. Model behaviour	93.5
6.	20. Functional and design constraint specification for the system	97.0
7.	23. Description and model of actual decision logic	140.0
8.	22. Creating a term dictionary	140.5
9.	31. Creating user stories	142.5
10.	17. Develop scenarios with “on-site customer”	165.0
11.	30. Modelling “As-Is” and “To-Be” models	168.5
12.	18. Definition of at least one KPI (Key Performance Indicator) for every requirement	177.0
13.	11. Modelling requirements for automatising decision-making	179.5

Results suggest that requirements are specified, documented and system structure is often modelled. Modelling system structure can be done using Unified Modelling Language (UML) class diagrams [20], ER diagrams [21], Business Process Modelling Notation (BPMN) using conversation diagram and Decision Model and Notation (DMN) using decision requirements diagram and decision requirements graph [22]. In order to model system behaviour UML activity diagram [23], sequence diagram [24], state diagram [20], [22] and communication diagram [20], as well as Data Flow Diagram (DFD) [21] and BPMN process, collaboration and choreography diagrams can be used [22]. User stories, according to the experts, are not as an important activity as creating models for system structure and behaviour or documenting requirements in general. It is possible that part of expert group has experience or is currently working with the government sector, which could explain why user stories are

not frequently used as government projects, which often mean highly detailed software requirements specification. One of the activities, which are rated as least important is definition of at least one KPI for every requirement. This correlates with previously established fact that the experts regard elicitation of non-functional requirements as less important because definition of KPI is especially important to non-functional requirements since it serves as a measurement, which can be used to determine if the requirement is met by both development team and client [23]. Activity rated as least important in this group by the experts is modelling requirements for automatising decision-making, which has been suggested in order to present human decision-making and model requirements for automated decision-making using DMN [22]. The first formal version of DMN was released by Object Management Group (OMG) in September 2015 [25], [22]. Therefore, it is possible that it has not gained popularity yet.

List of third group activities can be seen in Table [15].

TABLE V

THE THIRD GROUP SORTED BY IMPORTANCE

	The third group (K = 0.62)	Importance of activity (G_j)
1.	34. Using tools for requirements elicitation, analysis and specification	35.0
2.	25. Requirements validation	44.0
3.	33. Creating demos in a form of black and white screens, mockup sites	57.5
4.	24. Cataloguing suggestions and user remarks	59.0
5.	26. Ensuring quality (requirements non-redundant, concrete, understandable, clear, verifiable, feasible and unambiguous) and eliminating identified flaws	71.0
6.	27. Written requirements semantic and syntactic analysis	99.0
7.	32. Relative customer/stakeholder importance according to their level of interest and influence assessment	110.5

In this activity group, the most important activities are the use of tools for requirements elicitation, analysis and specification, requirements validation, creating demos and cataloguing suggestions and user remarks. One can conclude that quality of requirements is evaluated less frequently than requirements validation is performed. Less frequent activity is written requirements semantic and syntactic analysis, which has been a research topic for many authors who state that requirements written in natural language have different kinds of ambiguity [12] and have quality defects [11]. Creating demos in a form of black and white screens, mockup sites have been established as one of more important activities and have been suggested by authors [21], [26] as an activity, which helps ensure that stakeholders are satisfied with the outcome of project as most of the time stakeholders have little knowledge about information technology and might have different lexical perception [26]. It is possible that this activity has not been rated higher because not all projects require such an activity, or the demos, visuals and mockup sites are created by a designer or front-end developer. Activity rated as least important in this group is relative customer/stakeholder importance according to their level of interest and influence assessment, which has been suggested for creating Client Impact Map, which is further used in a Kanban board [4].

Seven of activities had options, which specified how exactly they were performed [15]. For the following analysis, all responses were considered as they might give more insight into the way the experts perform activities.

- Conducting interviews

Options to choose were as follows: Users, Stakeholders. From Fig. 1, it can be concluded that most of the time stakeholders are the ones who are interviewed. Users are interviewed almost as frequently as stakeholders.

- Stakeholder and/or user identification

Options were: stakeholder identification, user identification, key stakeholder identification, project sponsor identification. Figure 2 shows that most frequently users are identified following stakeholders and key stakeholders. Project sponsors that can be the same persons as key stakeholders are identified less often.

- Determining how and which data are used in the process

Participants were asked what data were determined: Documents, Data, IT systems, Services.

Figure 3 suggests that most of the time data to be used and relevant IT systems are identified, closely followed by services and relevant documents.

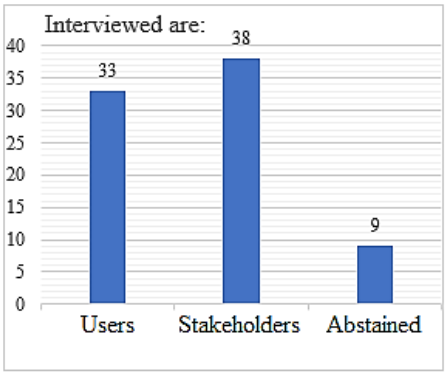

Fig. 1. Conducting interviews.

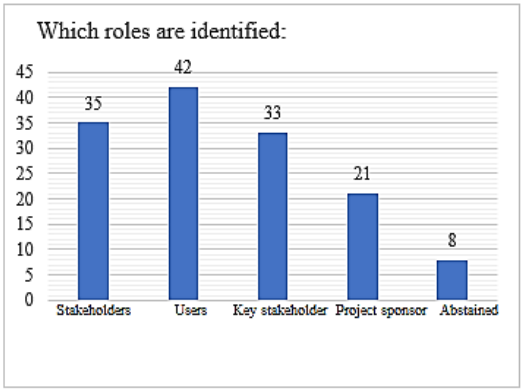

Fig. 2. Stakeholder and/or user identification.

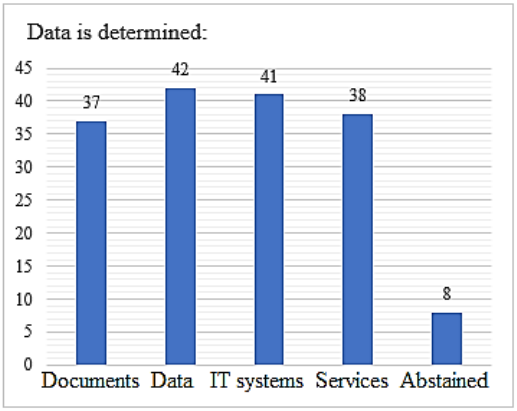

Fig. 3. Determining how and which data are used in the process.

- Requirements documentation

Experts were asked to specify how requirements were documented. The options were as follows: software requirements specification(1), graphical user interface description(2), unexpected input/situations(3), user stories(4). As can be seen in Fig. 4, requirements are generally documented in a form of software requirements(1) specification and description of user interface(2). Documentation in form of user stories(4) is less common. Additionally, experts provided with their input on this topic and wrote down how they documented requirements. Responses were: UML and other visual models, BDD (Behaviour Driven Development) scenarios, wireframes, prototypes, backlog, process models, design views, impact mappings, software design description, other formats, which are not necessarily standardised but are more easily understood by the client [15]. This feedback and input can be viewed as positive attitude towards survey and willingness of experts to share their insight. It also shows what industry experts use and can be further investigated why, for example, BDD scenarios, which belong to Agile methodologies [27], for some seem to be useful and what challenges this methodology poses.

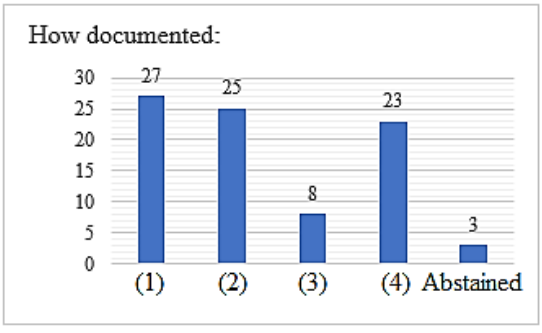

Fig. 4. Requirements documentation.

- Modelling system structure

Experts were asked about structure modelling: conceptual model, class diagram, ER diagram or other. As Fig. 5 indicates, structure mostly is modelled using conceptual models, closely followed by ER diagrams and finally – class diagrams. Once again, experts provided with their input and wrote down how they perceived and modelled system structure. Responses were: API-first approach, functional model, business process modelling, and deployment view [15]. The provided responses were further investigated by finding most probable explanation as to what exactly was meant. It was concluded that API-first approach most probably referred to the creation of a low fidelity prototype, which visualised system structure [15]. Functional model could be a reference to multiple different models but it was concluded that it could be part of formalised process description model as it has a functional model mentioned in its description [28]. Response business process modelling could possibly refer to BPMN mentioned before or other notations, which supported modelling of business processes. Deployment view could indicate the use of UML deployment diagram or diagram specified by RUP (Rational Unified Process), which described physical distribution of the system [15].

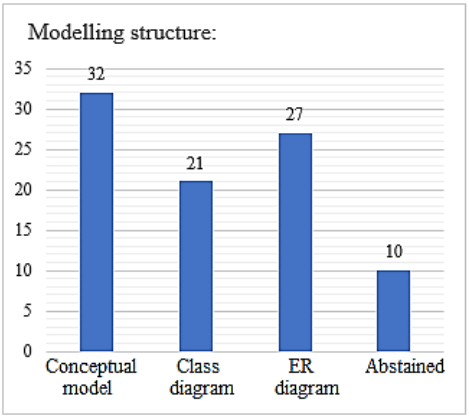

Fig. 5. Modelling system structure.

- Modelling behaviour

Survey participants were asked about the way behaviour was modelled. The options were as follows: sequence diagram(1), business process diagram(2), use case diagram(3), activity diagram(4), data flow diagram(5) or other. Figure 6 presents results, which indicate that system behaviour is most commonly modelled with business process diagrams and data flow diagrams. Use case diagrams, sequence diagrams, and activity diagrams are less commonly used. The authors have suggested that data flow diagrams are easily perceived by clients [21], which may be the reason the experts use this kind of diagram.

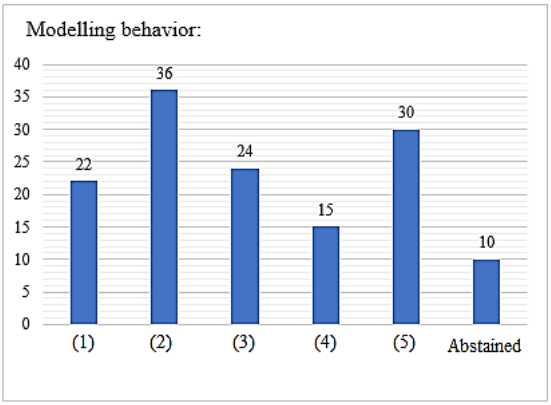

Fig. 6. Modelling system behavior.

Although the experts had an opportunity to write down other ways they were modelling behaviour, they did not give any additional answers. The reason might be that the experts rated the modelling behaviour as a less important activity compared to the modelling structure.

- Creating user stories

Participants of the survey were asked when/how user stories were created. Options were: large user stories – "Epics"(1), using template: "As a <type of user>, I want <goal>", [so that<reason/benefit>](2), converting use cases to user stories(3) or other. Figure 7 shows that even though only 24 experts responded that they modelled system behaviour with use case diagrams, 34 experts responded that they converted use cases to user stories (see Fig. 6). This can indicate that the experts do not see use cases as modelling system behaviour and still use them or they model system behaviour using use cases rarely but when they do, they further convert them into user stories. Another possibility is that the experts create use cases or something similar to them but do not create use case diagrams. One expert provided additional response: "User stories as well result from raised defects (during testing/using the system)" [15]. It indicates that there are user stories, which originate from defects and flaws found while testing or while system is already operating. It might be interesting to investigate what percentage of user stories originate from bugs or unexpected behaviour as it might reveal new information about how practitioners use this form of requirements and whether there are cases where a new user story is created because the previous one had quality defects or was incorrect.

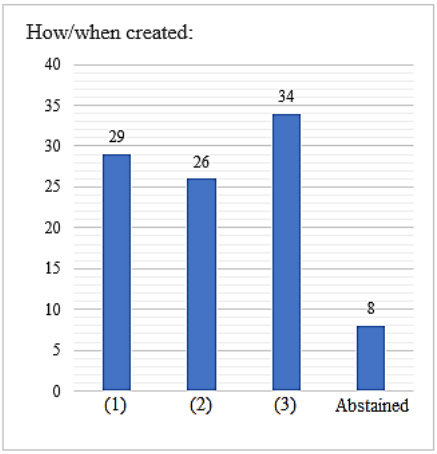

Fig. 7. Creating user stories.

- Using tools for requirements elicitation, analysis and specification

Experts were asked to rate importance of using tools in the system analysis and were provided with options: Microsoft Visio, Visual Studio, Enterprise Architect, IBM Rational DOORS, Medelio, ModelMaker, Visual-paradigm for UML or other. Figure 8 presents that 85.71 % of experts use Microsoft Visio, 34.6 % use Enterprise Architect, 26.53 % use Visual Studio, which until 2015 supported UML modelling, and 2.04 % use Visual-paradigm for UML [15].

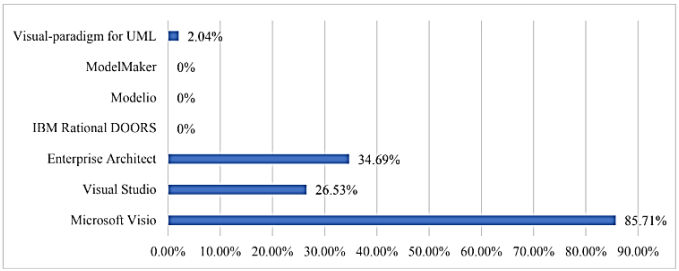

Fig. 8. Using tools for the system analysis.

Participants of the survey also listed tools they used for the system analysis. As shown in Fig. 9, there were many tools named by the experts. Balsamiq, InVision, moqups.com, Axure, XMind are tools used to create sketches, wireframes and mockups [15] and, therefore, can be used to visualise requirements. In previous sections, it has been established that activity "creating demos in a form of black and white screens, mockup sites" is a recognised and performed activity by the experts (Table V) and their responses to this question about tools indicates that these tools help in this activity. One of the experts provided the answer – pencil, which can mean both software and a physical tool for drawing. Sketches, ideas could be drawn by hand, for example, during interviews. Responses also contained Microsoft Word, Excel and PowerPoint [15], which can be used to document and present requirements. Altova Studio and XML Spy are created by company Altova and can be used to create UML diagrams, work with XML (Extensible Markup Language), generate code in Java, C#, C++. They can be integrated with development environments Eclipse and Visual Studio [15]. Skype as a tool for communication and Snag-it, which is used to screen capture and DB tools (which could possibly refer to many tools) were mentioned in the answers [15]. It can be concluded that the experts use more than one tool for elicitation, analysis and specification of requirements.

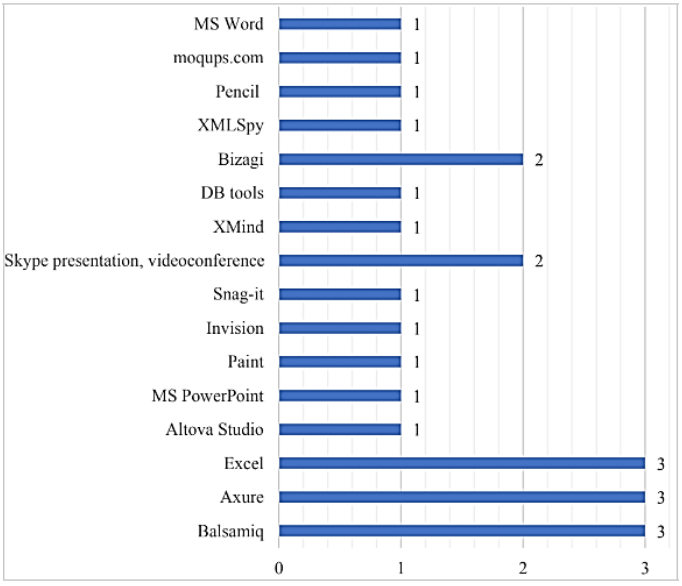

Fig. 9. Tools mentioned by experts in their responses.

IV. Related Work

Research on expert opinions on topics related to software development, system analysis has provided an insight into industry practices. A very similar research was conducted in a form of Master Thesis by S. Ciemina in 2010 [29]. S. Ciemina defined activities based on SWEBOK (Software Engineering Body of Knowledge), selected a smaller set of activities, chose 10 relevant experts with similar competence and used Delphi method to evaluate results [29]. Similarly to the results of the present research, S. Ciemina concluded that experts deemed functional requirements elicitation and requirements elicitation in general as one of the most important activities [29]. On contrary to the present research, S. Ciemina concluded that non-functional requirements ranked 5th out of eleven overall activities [29]. Conclusion might be that over the years attitude towards or perception of elicitation of non-functional requirements has changed. It is possible that perception of non-functional requirements has changed, or it is done by other team members with respective knowledge necessary to elicit these requirements. In order to make more precise conclusions, further research should be conducted. Another difference lies in the classification of activities. S. Ciemina classified conceptual modelling as requirements specification activity [29]. Its equivalent in the present research is requirements specification activity – system structure modelling [15] as system structure modelling can be performed using various models, including, conceptual [15].

Over the years, M. Kassab has consistently conducted research on topics related to requirements engineering. M. Kassab has concluded that the most popular requirements elicitation activities are user stories, brainstorming and interviews [30]. In the present research, interviews are recognised as a relatively important activity, and it might be concluded that results are similar to that of M. Kassab's research. Brainstorming is a similar activity to an interview, but it has a different level of formality and format. M. Kassab's research suggests that requirements are mostly expressed in natural language [30], [31], which corresponds to results of the present research as the experts have recognised requirements documentation, often in form of software requirements specification, as important activities. M. Kassab's research also indicates that automated requirements inspection tools are unpopular among the respondents of the survey [30], which conforms to a conclusion the present research has made – written requirements semantic and syntactic analysis is also unpopular among experts. M. Kassab has also reported on the use of tools in requirements engineering process – 64 % [31], which is lower than the results of the present research – 85.71 %. The main difference between the present research and M. Kassab's work is the respondents and their relevant competence. As stated in previous sections, the research only analysed responses of experts involved in requirements elicitation, analysis and documentation. However, M. Kassab's research involved software developers and testers and there was not any acknowledgement that these experts, while being possibly competent in the theory of requirements engineering practices, were involved in this process.

V. Conclusion and Future Work

In the paper, the authors have presented their results of the survey on activities in requirements elicitation, analysis and specification process. Survey participants were relevant experts with similar competence and their responses were evaluated using Delphi method and activities were divided into three groups and ranked according to their rated importance. The first group contained activities mainly related to requirements elicitation, the second group was related to requirements specification and the third – to requirements validation and other activities. Survey participants were responsive enough to provide their own answers when such an opportunity was given.

Requirements elicitation was interpreted as functional requirements elicitation by this expert group. Non-functional requirements elicitation was ranked as one of the least important activities and it possibly could have many explanations. One of them might be that participants work with well-established platforms, which already have non-functional characteristics developed, so a targeted elicitation of non-functional requirements is not necessary. If participants use Agile methodologies, it might be the cause of neglect of non-functional requirement elicitation as it is an established issue by researchers. Another possible explanation is that non-functional requirements are elicited by other team members with respective knowledge of security, usability, performance and other aspects, for example, developers, architects.

Use of surveys for requirements elicitation process was identified as rarely performed. It might be due to the use of Agile methodologies, which are focused on direct communication and collaboration. More frequently interviews are conducted.

It can be concluded that optimisation of business processes is not a frequently set objective for projects due to the fact that using business process models for requirements elicitation and identification of business rules was ranked as not an important activity. It is possible that some clients have not explicitly defined business processes and business rules.

Requirements specification in form of documents and modelling system structure are important activities according to the experts. System boundary and scope analysis and modelling systems behaviour are also performed activities.

Developing scenarios with "on-site customer" was ranked as less important than other activities, which could mean that not all clients were ready to contribute significant amount of stakeholder's time for this purpose. This activity can benefit outcome of the project and improve collaboration and communication among participants of the project. On the other hand, it can be viewed as an increase in costs of project by the client. Another activity that was ranked as less important than others was definition of at least one KPI for every requirement, which was especially important for non-functional requirements. As it was concluded – non-functional requirements were ranked as one of the least important activities; therefore, it was not surprising that this activity was also ranked as less important.

Use of tools was recognised as an important part of system analysis process, and the feedback provided by the experts

could give an insight into what practitioners in industry actually used.

Overall, the present research provides an insight into how industry experts perform requirements elicitation, analysis and specification, what tools contribute to this process. These data can be used to conduct further investigation on industry practices related to requirements engineering. The presented data can also provide information to other practitioners on how their industry colleagues approach tasks related to requirements engineering, what tools others find beneficial for this process. The information can contribute to solving issues that other industry practitioners have already solved.

The authors intend to further investigate the obtained results in order to gain more insight into the use of activities, tools and issues practitioners encounter in their work.

REFERENCES

[1] A. Batool, Y. H. Motla, Bushra, S. Asghar, M. Riaz, M. Mukhtar, and M. Ahmed, "Comparative Study of Traditional Requirement Engineering and Agile Requirement Engineering" in *International Conference on Advanced Communication Technology (ICACT)*, 2013, pp. 1006–1014.

[2] K. S. Rubin. *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* USA: Addison-Wesley Professional, 2012. 500 p. ISBN 978-0137043293.

[3] K. Beck, M. Beedle, A. van Bennekum, et al., *Manifesto for Agile Software Development*, 2001. [Online]. Available: http://agilemanifesto.org/

[4] W. Shim and S. W. Lee, "An Agile Approach for Managing Requirements to Improve Learning and Adaptability," in *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, 2017, pp. 435–438. https://doi.org/10.1109/REW.2017.46

[5] N. Bik, G. Lucassen, and S. Brinkkemper, "A Reference Method for User Story Requirements in Agile Systems Development," in *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, 2017, pp. 268–275. https://doi.org/10.1109/REW.2017.83

[6] K. Pohl and C. Rupp, *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam – Foundation Level – IREB compliant*, 2nd ed. USA: Rocky Nook Inc., 2015. 184 p. ISBN 978-1937538774.

[7] P. Kummler, L. Vernisse, and H. Fromm, "How Good Are My Requirements?: A New Perspective on the Quality Measurement of Textual Requirements," in *11th International Conference on the Quality of Information and Communications Technology (QUATIC)*, 2018, pp. 156–159. https://doi.org/10.1109/QUATIC.2018.00031

[8] R. R. Maiti and F. J. Mitropoulos, "Prioritizing Non-Functional Requirements in Agile Software Engineering" in *Proceedings of the SouthEast Conference (ACM SE '17)*, 2017, pp. 212–214. https://doi.org/10.1145/3077286.3077565

[9] J. Slankas and L. Williams, "Automated Extraction of Non-functional Requirements in Available Documentation," in *1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE)*, 2013, pp. 9–16. https://doi.org/10.1109/NAturaLiSE.2013.6611715

[10] S. Anwer, L. Wen, and Z. Wang, "A Systematic Approach for Identifying Requirement Change Management Challenges: Preliminary Results" in *Proceedings of the Evaluation and Assessment on Software Engineering (EASE '19)*, 2019, pp. 230–235. https://doi.org/10.1145/3319008.3319031

[11] F. Dalpiaz and S. Brinkkemper, "Agile Requirements Engineering With User Stories," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, Banff, AB, 2018, pp. 506–507. https://doi.org/10.1109/RE.2018.00075

[12] A. O. J. Sabriye and W. M. N. W. Zainon, "A Framework for Detecting Ambiguity in Software Requirement Specification" in *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, 2017, pp. 209–2013. https://doi.org/10.1109/ICITECH.2017.8080002

[13] L. Leimane and O. Ņikiforova, "Mapping of Activities for Object-Oriented System Analysis", *Applied Computer Systems*, vol. 23, iss. 1, pp. 5–11, May 2018. https://doi.org/10.2478/acss-2018-0001

[14] O. Ņikiforova and U. Sukovskis, "Framework for Comparison of System Modelling Tools," in *Proceedings of Fifth IEEE International Baltic Workshop on DB and IS, BalticDB&IS'*, vol. 1, 2002, pp. 63–70.

[15] L. Leimane, "Research on Usage of Object-Oriented System Analysis Methods for Software Requirement Determination," Master thesis, Riga Technical University, 2018, 90 p.

[16] K. K. Lilja, K. Laakso, and J. Palomaki, "Using the Delphi Method," in *2011 Proceedings of PICMET '11: Technology Management in the Energy Smart World (PICMET)*, 2011.

[17] H. B. Triantono, M. B. Susetyarto, "Technical Use of Analytical Hierarchy Process and Delphi Method in Determining Terminal Location Multi Function of Merak Port," in *2017 International Conference on Information Management and Technology (ICIMTech)*, 2018, pp. 350–355. https://doi.org/10.1109/ICIMTech.2017.8273564

[18] A. Silva, P. R. Pinheiro, A. Albuquerque, and J. Barroso, "Evaluation of an Approach to Define Elicitation Guides of Non-Functional Requirements," *IET Software*, vol. 11, iss. 5, pp. 221–228, Oct. 2017. https://doi.org/10.1049/iet-sen.2016.0302

[19] D. Kung, J. Lei, "An Object-Oriented Analysis and Design Environment," in *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, 2016, pp. 91–100. https://doi.org/10.1109/CSEET.2016.20

[20] Q. Wei, G. Danwei, X. Yaohong, F. Jingtao et al., "Research on Software Development Process Conjunction of Scrum and UML modeling," in *2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, 2014, pp. 978–982. https://doi.org/10.1109/IMCCC.2014.206

[21] V. I. Abdulaev, *Programmnaja inzhenerija: uchebnoe posobie*, Ch. 1. Proektirovanie sistem. Russia: PGTU, 2016.

[22] K. Kluza, P. Wiśniewski, K. Jobczyk, A. Ligeza, "Comparison of Selected Modeling Notations for Process, Decision and System Modeling," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, pp. 1095–1098. https://doi.org/10.15439/2017F454

[23] S. Dragicevic, S. Celar, L. Novak, "Use of Method for Elicitation, Documentation and Validation of Software User Requirements (MEDoV) in Agile Software Development Projects," in *2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, 2014, pp. 65–70. https://doi.org/10.1109/CICSyN.2014.27

[24] A. Zeaaraoui, Z. Bougroun, M. G. Belkasmi, T. Bouchentouf, "User Stories Template for Object-Oriented Applications," in *2013 Third International Conference on Innovative Computing Technology (INTECH)*, 2013, pp. 407–409. https://doi.org/10.1109/INTECH.2013.6653681

[25] "About the Decision Model and Notation specification version 1.0," Object Management Group, 2015. [Online]. Available: https://www.omg.org/spec/DMN/1.0

[26] A. P. Murray, *The Complete Software Project Manager: Mastering Technology From Planning to Launch and Beyond*. USA: John Wiley & Sons, Incorporated, 2016. https://doi.org/10.1002/9781119220015

[27] L. P. Binamungu, S. M. Embury, N. Konstantinou, "Maintaining Behaviour Driven Development Specifications: Challenges and Opportunities," in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2018. pp. 175–184. https://doi.org/10.1109/SANER.2018.8330207

[28] A. Scholz, C. Hildebrandt, A. Fay, "Functional Modelling in Production Engineering Workflows," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 2018. pp. 695–700. https://doi.org/10.1109/COASE.2017.8256184

[29] S. Ciemiņa, "Research of Object Oriented System Analysis Activities," Master thesis, Riga Technical University, 2010, 65 p.

[30] M. Kassab, "An Empirical Study on the Requirements Engineering Practices for Agile Software Development," in *2014 40th Euromicro Conference on Software Engineering and Advanced Applications*, 2014, pp. 254–261. https://doi.org/10.1109/SEAA.2014.77

[31] M. Kassab, "The Changing Landscape of Requirements Engineering Practices over the Past Decade," in *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2015, pp. 1–8. https://doi.org/10.1109/EmpiRE.2015.7431299

Laima Leimane received the Master degree in computer systems from Riga Technical University, Latvia, in 2018. She is presently the first-year Doctoral student at the Department of Applied Computer Science, Riga Technical University.
E-mail: laima.leimane@edu.rtu.lv

Oksana Nikiforova received the Doctoral degree in information technologies (system analysis, modelling and design) from Riga Technical University, Latvia, in 2001.
She is presently a Professor at the Department of Applied Computer Science, Riga Technical University, where she has been working since 1997. Her current research interests include object-oriented system analysis, design and modelling, especially the issues in model driven software development.
E-mail: oksana.nikiforova@rtu.lv
ORCID iD: https://orcid.org/0000-0001-7983-3088