An Empirical Evaluation about Using Models to Improve Preliminary Safety Analysis

Jéssyka Vilela

Universidade Federal de Pernambuco (UFPE)
Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária (Campus Recife), CEP: 50.740-560 - Recife - PE

jffv@cin.ufpe.br

Abstract. Context*: Safety analysis is an activity of fundamental importance in the development of safety-critical systems (SCS) to ensure that hazardous situations are properly found and mitigated. Such analysis is performed after a system requirements specification is available. Therefore, it is then worthwhile to investigate specification techniques to detect their strengths and weaknesses with respect to discovering hazards early in the development process.* ***Objective****: In this paper, we investigate similarities and differences in the results of a preliminary safety analysis from requirements specified using models in Business Process Modeling Notation (BPMN) and Textual Use Cases (TUC).* ***Method****: We adopted a controlled experiment as research method using computer engineering students as subjects.* ***Results****: The subjects of BPMN group found more accidents, hazards as well as more causes of hazards. Moreover, they have a higher preference for the template used for safety analysis documentation.* ***Conclusions****: The use of BPMN to represent the interactions among actors in a system probably lead to the discovery of more accidents and hazards, but more experiments are necessary to test this hypothesis since the results are not statistically significant.*

1. Introduction

Many systems currently used in human daily life activities have safety-related implications (LEVESON, 2011) (KAINDL, POPP and RANEBURGER, 2015). Because of such hazardous situations, requirements-related problems have been recognized as the main cause of many accidents (Leveson, 2011). To mitigate such problems, the literature (Vilela et al., 2017) (Leveson, 2011) reports and argues that preliminary safety analysis (PSA) should be conducted early in the development process. In PSA, engineers review the specification looking for hazards and their causes that could lead to accidents or safety incidents.

Considering that systems are becoming ever more safety-critical, it is important that software designers, developers and testers can contribute to hazard identification. Hence, it is necessary to evaluate if different specification techniques lead to either the identification of more hazards or to identify some specific category of hazards and accidents. Researchers have investigated the languages that lead to uncover more hazards (Stålhane and Sindre, 2014) (Stålhane Sindre and Du Bousquet, 2010) and to specify the results of safety analysis.

In this paper, we conducted a controlled experiment with 12 computer engineering students, same strategy of previous works (Stålhane and Sindre, 2014) (Stålhane Sindre

and Du Bousquet, 2010), to evaluate two requirements techniques (textual use cases and BPMN). We investigated the languages' strengths and weaknesses with respect to finding and documenting hazards. BPMN allows to represent the interactions among actors (lanes), their actions (tasks), and events that triggers and constrain the tasks. This language is typically used for business process modeling, however, use cases may also be represented using this notation.

This paper is organized as follows. We provide a brief overview of safety analysis concepts and related work in Section 2. The experiment design is explained in Section 3. In Section 4, we describe the results. In Section 5, we discuss some threats to validity. Our conclusions as well as further research are presented in Section 6.

2. Background and Related Work

Safety-critical systems consist of a set of hardware, software, process, data and people whose failure could result in accidents that cause damage to the environment, financial losses, injury to people and loss of lives (LEVESON, 2011). Hence, the specification of such systems should be properly conducted (VILELA et al., 2017).

In PSA, accidents, hazards and their causes are determined. An accident is an undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss (Leveson, 2011) (including loss of human life or injury, property damage, environmental pollution, and so on). A hazard is a system state or set of conditions that, together with a set of worst-case environmental conditions, will lead to an accident (loss) (LEVESON, 2011). A cause of hazard is a reason that produces hazard as effect. They occur due to environmental hazard, procedural hazard, interface hazard, human factor or system cause (LEVESON, 2011). For example, in a train control system hazard may be a fault in the communication between the train and the operation room.

Controlled experiments have been used to access PSA based on different requirements specification languages (STÅLHANE and SINDRE, 2014) (STÅLHANE SINDRE and DU BOUSQUET, 2010). The experiments have been conducted to investigate the strengths and weaknesses of requirements techniques with respect to uncovering hazards. Stålhane and Sindre compared safety analysis performed on system diagrams and textual use cases (Stålhane and Sindre, 2014); and, in a subsequent work, Stålhane, Sindre, and Bousquet conducted an experiment using sequence diagrams and textual use cases (STÅLHANE SINDRE and DU BOUSQUET, 2010). Nevertheless, PSA from BPMN has not been evaluated. Our investigation complements above previous works regarding the determination of which RE language contributes to the identification of more hazards, allowing them to be mitigated and, consequently, to develop safer systems from the beginning.

3. Experiment Design

We followed the well-established framework proposed by Wohlin et al. (2012) for performing experiments in software engineering to ensure a successful experiment (Vilela et al., 2016). In this study, we intend to answer the following research questions:

RQ1: Does one of the two investigated languages (TUC and BPMN) better support hazard analysis?

RQ2: Do the investigated languages have different effectiveness for different types of hazards, or is one method uniformly better than the other?

The goal of the experiment is summarized in Table 1. The subjects were 12 third year computer engineering students. We have used convenience sampling and have split them into two groups (6 in the TUC group and 6 in the BPMN group) randomly.

Table 1. Goal of the experiment.

Analyze	Preliminary safety analysis results based on TUC and BPMN
For the purpose of	characterization
With respect to	Time spent on the analysis, number of hazards in the system, number of accidents, hazards and their causes found in the functionality.
From the point of view of	non-experts
In the context of	students of a computer engineering undergraduate course of the same period with no previous experience in requirements engineering and safety analysis enrolled in an embedded software engineering discipline performing the safety analysis of a train control system.

The students were enrolled in an embedded software engineering course and, before the experiment, we provided training to subjects about the main safety analysis concepts and safety analysis techniques in 4 hours, and classes about Use Cases and BPMN of 4 hours each to both groups. We did not perform a pilot study before the experiment.

Our **Independent variables** were the use of TUC or BPMN to represent the functionality to be analyzed and the **Dependent variables** were time spent on the analysis, number of accidents, hazards and their causes found related to the analyzed functionality. The main hypotheses are the null hypotheses that states there is no difference between performing safety analysis from TUC or BPMN. Table 2 describes the null and alternative hypotheses of this experiment.

Table 2. Null and alternative hypothesis.

Null hypothesis	**Alternative hypothesis**	**Alternative hypothesis**
H01: TimeTUC = TimeBPMN Group	H11: TimeTUC > TimeBPMN Group	H12: TimeTUC < TimeBPMN Group
H02:#AccidentsTUC= AccidentsBPMN	H13: #AccidentsTUC > #AccidentsBPMN	H14: #AccidentsTUC < #AccidentsBPMN
H03: #HazardsTUC = #HazardsBPMN	H15: #HazardsTUC > #HazardsBPMN	H16: #HazardsTUC < #HazardsBPMN
H04: #CausesTUC = #CausesBPMN	H17: #CausesTUC > #CausesBPMN	H18: #CausesTUC < #CausesBPMN

We compared two treatments: the functionality specified in TUC and in BPMN. Therefore, the design of our experiment is classified as one factor with two treatments being of the type completely randomized design (WOHLIN et al., 2012). All subjects analyzed the same simple train control system depicted in Figure 1.

The TUC group (group 1) received the system diagram and the TUC of Table 3 (the control operator doing train scheduling) both elaborated by Stålhane and Sindre (2014). The BPMN group (group 2) used the same system diagram and the same case specified in BPMN of Figure 2.

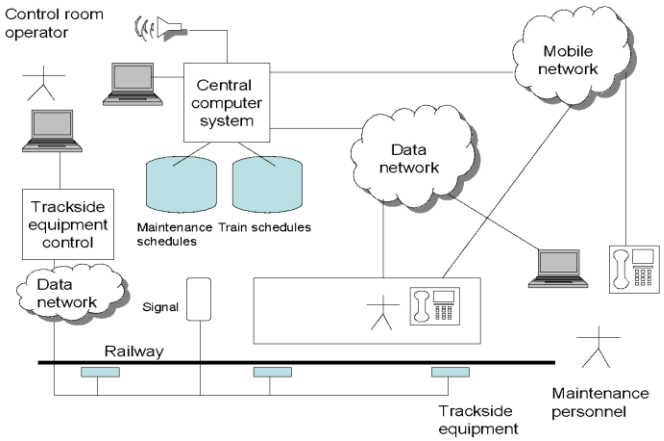

Figure 1. System diagram of a train control system (Stålhane and Sindre, 2014).

Table 3. Use case for (Re) Define train schedule (Stålhane and Sindre, 2014).

Name	(Re) Define train schedule
ID	FR01
Actors	Control Room Operator, Train onboard computer
Main path	1. User requests to enter schedule information. 2. System shows the scheduling screen. 3. User enters the schedule (train-ID, start and stop place and time, as well as timing for intermediate points). 4. System checks that the schedule does not conflict with other existing schedules; display entered schedule for confirmation. 5. User confirms schedule. 6. System sends scheduling to onboard train computer. 7. The train onboard computer receives the instructions. 8. The onboard computer of the train performs the scheduling as registered.
Alternative Path	The request is to edit an existing schedule: 2.1 The system shows the schedule. 3.1 The operator changes some information in the schedule, and then the use case proceeds as normal.
Exceptions	4. Schedule conflicts with another scheduled train or maintenance task. 5. Operator must decide whether to change the schedule or alternatively to reschedule also the other train/event. 5.1 User changes the schedule. 5.2 User changes the schedule of another train.

The experiment was carried out June, 2017 in a computer laboratory with all students at the same time with each student using an individual computer. The time available for conducting the experiment was 2 hours and the subjects were not graded based on their performance. We asked them to perform the following activities:

- To write down the start and end time of the experiment.
- To identify the accidents and incidents that may occur due to the use of the functionality *(Re) Define train schedule* specified in TUC and BPMN as well as possible damages.
- To identify the hazards related to each accident and incident previously listed.
- To identify the causes of each identified hazard.

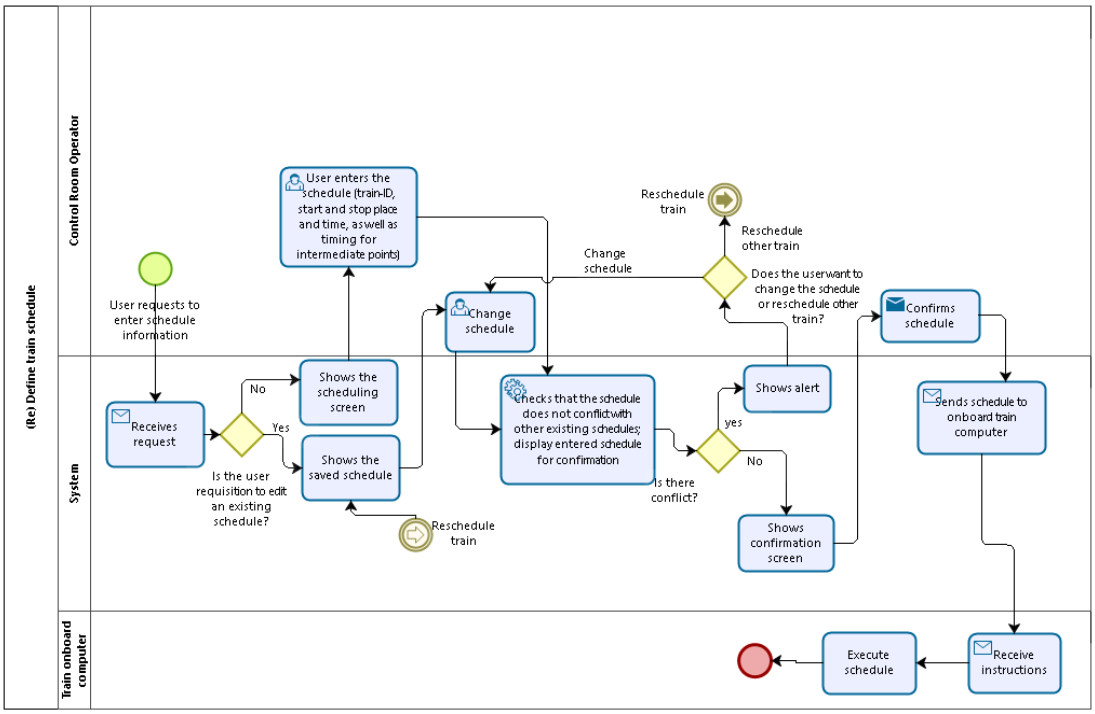

Figure 2. BPMN for (Re) Define train schedule.

In the aforementioned activities, each subject documented the identified hazards using an editable version of the template used for safety analysis documentation (see Table 4).

Table 4. Template for documenting the accidents, hazards and their causes.

Functionality: ***(Re) Define train schedule***							
Code	***Accident/Incident***	***Harm***	***Harm Type***	***Hazard Code***	***Hazard Description***	***Cause***	***Cause Type***

4. Results and discussion

Both TUC and BPMN are used at the RE stage of system development and allow to represent the same information (a sequence of steps to perform an activity). Considering the differences of representation types (textual x graphical), our goal was to analyze if there are differences in the number of accidents and hazards identified by both groups.

We requested the subjects to identify the accidents and incidents that may occur due to the use of the functionality *(Re) Define train schedule*. We did not prepare a document with the description of a set of potential accidents and hazards and hazard causes in the context of the given functionality and used this document to evaluate the results obtained by the subjects.

We observed that the subjects reported accidents that, although may happen in the system, they were not related to the functionality provided for analysis. Examples of such accidents were Collision with wild animals, Breaking rails, and Leakage of fuel. The

average of the number of accidents of TUC group was higher (Figure 3a), but when we consider only the functionality-related accidents (Figure 3b), the average of the accidents is less.

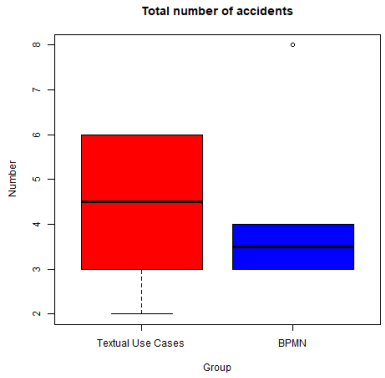

Figure 3(a). Boxplots of the total number of accidents reported by subjects.

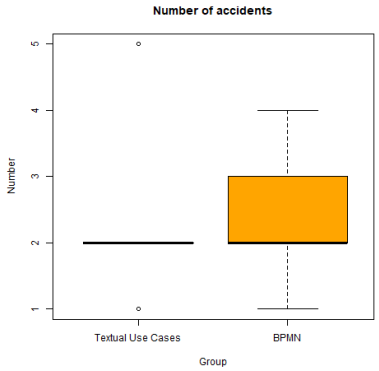

Figure 3(b). Boxplots of the variable number of functionality-related accidents.

In Table 5, we only show the accidents related to the functionality *(Re) Define train schedule* and the number of subjects that reported them.

Table 5. Average of number of subjects that reported accidents/incidents in both groups.

	Number of people reporting the accident		**Number of hazards reported per accident**		**Average of number of causes of the hazards of the functionality**	
Accident/Incident	**TUC**	**BPMN**	**TUC**	**BPMN**	**TUC**	**BPMN**
Train Collision	5	6	2.67	5.83	3.00	6.83
Train circulates on the wrong route	1	2	1.00	1.83	0.67	1.83
Train Collision with Infrastructure	1	1	1.00	0.50	0.67	0.33
Stolen data	1	0	0.17	0.00	0.17	0.00
Explosion of trains	1	0	0.83	0.00	0.50	0.00
A train collides with a maintenance crew	1	0	1.00	0.00	1.83	0.00
Train delay	0	2	0.00	1.17	0.00	1.0
Train derailment	2	4	0.17	2.5	0.33	1.33
Train does not move	2	0	0.17	0.00	0.33	0.00
Average			**7.00**	**10.43**	**7.5**	**11.33**

The next task consisted of identifying the hazards related to each accident previously specified. The boxplots of this dependent variable are presented in Figure 4(a). The average of hazards reported by both groups per accident/incident is described in Table 2.

We notice that BPMN group reported more hazards than TUC group. This result may occur to the fact that visual diagrams are better to analyze than information represented in natural language. Larkin and Simon (1987) have noticed similar result in their seminal paper providing critical insights into the potential benefits of diagrammatic representations over propositional or sentential representations.

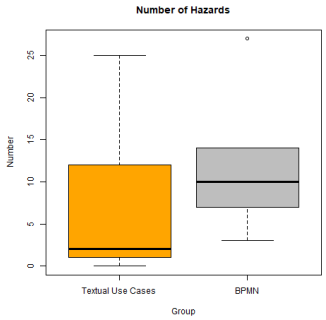

Figure 4(a). Boxplots of the variable number of hazards.

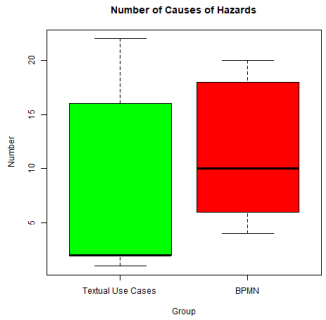

Figure 4(b). Boxplots of the variable causes of hazards.

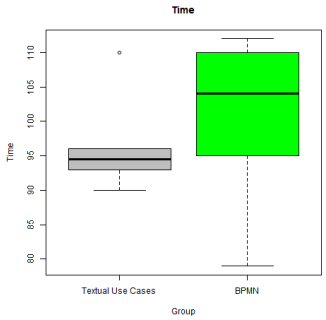

Figure 4(c). Boxplots of the variable time.

After the identification of the hazards, the next step in safety analysis is the documentation of their causes. For example, the accident *Train Collision* may occur to the hazard *Failure in checking stored schedule*. This hazard may have many causes such as *The system has not stored the schedule and a train will use that route*; or *The system does not perform a correct check due to a power failure or other technical problem*.

The distribution of this dependent variable is shown in Figure 4(b). Although the subject that found more hazards belong to the TUC group, the average of the group (7.5) was less than the BPMN group (11.33). A possible reason to why textual use cases group did not work well could be the confusion caused by the numbering scheme for the steps. This observation has also been found by Alspaugh et al. (2010). Moreover, experiments have also found diagrams superior for presenting procedural instructions (STÅLHANE, SINDRE and DU BOUSQUET, 2010).

The time spent by each subject to perform the safety analysis is presented in Figure 4(c). We have noticed that the time spent by BPMN group was slightly higher ($\mu = 100.67$ min) than the Textual Use Cases group ($\mu = 96.33$ min) with a small difference of 4.33 min (4.5%).

In this work, we used the Kolmogorov-Smirnov normality test whose results are listed in Table 6. Since the data are normally distributed with a confidence level of 95%, we applied parametric tests in order to perform the hypotheses testing. The results of this experiment, presented in Table 7, did not allow the rejection of our null hypothesis (all p-values are higher than 0.05). Hence, we cannot say the results are statically relevant requiring its replication with larger samples. However, this is a new contribution in the investigation of which requirements representation lead to better safety analysis results we compare two languages that have not been compared before.

After the experiment, the subjects answered a short questionnaire with 9 questions to evaluate the template proposed for safety analysis specification considering Perceived Ease of Use (PEOU), Perceived Usefulness (PU), and Intention to Use (ITU). The subjects' opinion of both groups using the five-point Likert scale are presented in Table

8. Accordingly, we noticed that the BPMN group that found more accidents, more hazards as well as more causes of hazards is the one that have the higher preference of the template.

Table 6. Results of normality statistical testing of investigated variables of TUC group.

Variable	Normality test	Statistical comparison
	TUC	BPMN
Time	Normal (p-value = 0.5717)	Normal (p-value = 0.9397)
Number of accidents	Normal (p-value = 0.2179)	Normal (p-value = 0.8103)
Number of hazards	Normal (p-value = 0.4148)	Normal (p-value = 0.9014)
Number of causes	Normal (p-value = 0.3107)	Normal (p-value = 0.8026)

Table 7. Results of hypothesis testing.

Hypothesis test	Result	Conclusion
H01 and H11	T-Test(p-value= 0.7624)	H01 not rejected
H01 and H12	T-Test(p-value= 0.2376)	H02 not rejected
H02 and H13	T-Test(p-value=0.555)	H03 not rejected
H02 and H14	T-Test(p-value=0.445)	H04 not rejected
H03 and H15	T-Test(p-value=0.8079)	H05 not rejected
H03 and H16	T-Test(p-value=0.1921)	H06 not rejected
H04 and H17	T-Test(p-value=0.7864)	H07 not rejected
H04 and H18	T-Test(p-value=0.2136)	H08 not rejected

Table 8. Post-Experiment Questionnaire.

	Textual Use Cases					BPMN				
Question	**TA (%)**	**PA (%)**	**N (%)**	**PD (%)**	**TD (%)**	**TA (%)**	**PA (%)**	**N (%)**	**PD (%)**	**TD (%)**
Q1: The presented template is easy to use.	33.33	0	16.67	16.67	0	33.33	66.67	0	0	0
Q2: The template is easy to understand.	16.67	33.33	16.67	16.67	0	33.33	50	16.67	0	0
Q3: The template made the safety analysis more systematic.	0	66.67	0	16.67	0	66.67	33.33	0	0	0
Q4: The template helped me to focus on the information that needs to be identified during the safety analysis.	16.67	50	0	16.67	16.67	66.67	33.33	0	0	0
Q5: If I need to identify safety threats in a future project, I would use the template.	0	50	0	16.67	0	33.33	50	16.67	0	0
Q6: I was never confused about how to the use the template.	0	0	0	33.33	0	33.33	16.67	16.67	16.67	16.67
Q7: The template worked well as a guide to safety analysis.	0	50	0	16.67	0	50	33.33	16.67	0	0
Q8: If working as a freelance consultant for a customer who needs help to find safety threats to his/her software, I would like to use the template in discussions with that customer.	0	33.33	16.67	16.67	0	33.33	50	16.67	0	0
Q9: The template contributes to the improvement of safety analysis of embedded systems.	33.33	16.67	0	16.67	0	33.33	50	16.67	0	0
Legend: TA=Totally Agree, PA=Partially Agree, N=Neutral, PD=Partially Disagree, TD=Totally Disagree										

We followed the validity categories proposed by Wohlin et al. (2012) to analyze the threats to validity of this experiment. *Internal Validity*: we tried to reduce selection bias by performing a random assignment of the subjects to the groups. Moreover, given that the experiment was performed in one day related to a domain that they had no contact before, we have mitigated the history and maturation effects by making observation at just one time. *Conclusion validity:* we tried to improve the reliability by providing the same training, with the same instructor for all subjects.

Construct Validity: since the subjects had no previous experience in requirements techniques before the course and all received the same training, we may conclude that the differences in the results are related to the technique they have used (TUC or BPMN). *External validity:* the context of our experiment is undergraduate students. It has been demonstrated that there is not necessarily much difference between students and professionals in many experimental settings (Svahnberg, Aurum and Wohlin, 2008). Although the results are limited by the narrow scope, the study design can guide other studies to compare requirements techniques and it can also support other kind of studies (Vilela et al., 2016).

5. Conclusions

In this paper, we reported the results of a controlled experiment conducted to investigate the safety analysis performed from TUC and BPMN. The subjects were 12 students of an undergraduate computer engineering course enrolled in an embedded software engineering discipline.

There were some categories in common in both groups such as Changes in the environment, Human Error, and Wrong Information. However, the TUC group considered more hazards related to Health Issues, while the BPMN group considered Climate changes and identified more hazard related to failure of equipment.

The BPMN group found more accidents (7.14% more), more hazards (69.04%) as well as more causes of hazards (51.11%). The differences in the results cannot stem from the presence or absence of system diagrams as the diagram was available in both sets of documentation. The results possible indicate that the BPMN graphical representation is better than textual use cases to represent the sequence of activities and interactions among actors. Such representation seems to contribute to better safety analysis of functionalities in which the interaction among actors and wrong sequence of activities cause accidents, but more experiments are necessary before making a final decision.

Human information processing is highly sensitive to the exact form in which information is presented to the senses (MOODY, 2009). Accordingly, the confusion caused by the numbering scheme for the steps (Stålhane and Sindre, 2014) could be the reason why textual use cases group did not work well as BPMN group. Moreover, experiments have also concluded that diagrams are superior for presenting procedural instructions (STÅLHANE, SINDRE, DU BOUSQUET, 2010)(ALSPAUGH et al., 2007).

The safety analysis based on BPMN diagrams is a novel contribution towards the determination of languages most adequate to specify requirements of a safety-critical system. In future work, we intend to do more experiments with larger samples to obtain statistically significant results. It would also be interesting to compare the techniques

using larger case studies, since controlled experiments necessarily limit the size of the tasks that can be performed, thus lacking the realism and complexity of information systems development projects.

ACKNOWLEDGMENTS

We want to thank the students that participated in the experiment and to Tor Stålhane and Lydie du Bousquet that made available the material of their experiments.

References

Vilela, Jéssyka; Castro, Jaelson; Martins, Luiz Eduardo G.; Gorschek, Tony. Integration between requirements engineering and safety analysis: A systematic literature review. Journal of Systems and Software, v. 125, pp. 68-92, 2017. DOI: https://doi.org/10.1016/j.jss.2016.11.031

Wohlin, Claes et al. Experimentation in software engineering. Springer Science & Business Media, 2012.

Stålhane T, Sindre G. An experimental comparison of system diagrams and textual use cases for the identification of safety hazards. In: International Journal of Information System Modeling and Design (IJISMD), 5 (1), 2014, pp. 1-24.

Leveson, N. Engineering a Safer World: Systems Thinking Applied to Safety. Mit Press, 2011.

H. Kaindl, R. Popp, D. Raneburger. Towards reuse in safety risk analysis based on product line requirements. In: 23rd International Requirements Engineering Conference (RE), 2015, pp. 241-246.

Stålhane T, Sindre G, Du Bousquet L. Comparing safety analysis based on sequence diagrams and textual use cases. In: Advanced Information Systems Engineering, 2010, pp. 165-179.

Alspaugh TA, Sim SE, Winbladh K, Leila MH, Ziv H, Richardson DJ. Clarity for stakeholders: Empirical evaluation of scenarioml, use cases, and sequence diagrams. In: International Workshop on Comparative Evaluation in Requirements Engineering, 2007, pp. 1-10.

Svahnberg M, Aurum A, Wohlin C (2008) Using students as subjects - an empirical evaluation. In: Proceedings of International Symposium on Empirical Software Engineering and Measurement, 2008, pp 288–290.

Larkin, J.H. and H.A. Simon, Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, 1987, pp. 65-99.

Vilela, Jéssyka; Castro, Jaelson; Pimentel, João. A systematic process for obtaining the behavior of context-sensitive systems. Journal of Software Engineering Research and Development, v. 4, n. 1, p. 2, 2016.

Moody D. The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. In: IEEE Transactions on Software Engineering. 2009 Nov; 35 (6), pp. 756-79.