2020 IEEE 28th International Requirements Engineering Conference (RE)

Inspectors Academy

Pedagogical Design for Requirements Inspection Training

Muneera Bano[1], Didar Zowghi[2], Alessio Ferrari[3], Paola Spoletini[4]
[1]Faculty of Science, Engineering and Built Environment, Deakin University, Australia
[2]Faculty of Engineering and IT, University of Technology Sydney, Australia
[3]ISTI-CNR, Pisa, Italy
[4]Kennesaw State University, USA
[1]muneera.bano@deakin.edu.au, [2]Didar.Zowghi@uts.edu.au, [3]alessio.ferrari@isti.cnr.it, [4]pspoleti@kennesaw.edu

***Abstract*— The core aim of requirements inspection is to ensure the high quality of already elicited requirements in the Software Requirements Specification. Teaching requirements inspection to novices is challenging, as inspecting requirements needs several skills as well as knowledge of the product and process that is hard to achieve in a classroom environment. Published studies about pedagogical design specifically for teaching requirements inspection are scarce. Our objective is to present the design and evaluation of a postgraduate course for requirements inspection training. We conducted an empirical study with 138 postgraduate students, teamed up in 34 groups to conduct requirements inspection. We performed qualitative analysis on the data collected from students' reflection reports to assess the effects of the pedagogical design in terms of benefits and challenges. We also quantitatively analyze the correlation between the students' performance in conducting inspections and their ability of writing specifications. From the analysis of students' reflections, several themes emerged such as their difficulty of working with limited information, but also revealed the benefits of learning teamwork and writing good requirements. This qualitative analysis also provides recommendations for improving the related activities. The results revealed a moderate positive correlation between the performance in writing specification and inspection.**

***Index Terms*—Requirements Inspection, Requirements Engineering Education and Training, Empirical Study**

I. Introduction

Since initially introduced to find defects in the source code [1], formal inspections have become one of the most common approaches for finding defects in software requirements specifications (SRS). Requirements inspections are an important part of the software development lifecycle (SDLC) to ensure high quality of requirements. It is estimated that software engineers spend around 80% of the total development time on testing and debugging, the majority of the bugs are attributed to early phases of Requirements Engineering (RE) [2]. It is therefore cost effective and easier to address the issues in requirements earlier rather than later in SDLC [3]. Empirical evidence collected from industrial projects have indicated that requirements inspection has a significant impact on the quality of requirements [4].

The fast-evolving software industry requires the educational institutions offering Software Engineering (SE) degrees to produce industry ready graduates. The educators are therefore redesigning their curricula to provide a simulated real-world environment within academia to prepare the students for the work environment. Training students to perform requirements inspection is challenging. When it comes to the empirical evidence for effective pedagogical designs for teaching students requirements inspection techniques, literature in Requirements Engineering Education and Training (REET) [5] is scarce.

Our motivation for this study stems from many years of academic experiences of teaching RE and observing how university students (both undergraduate and postgraduate levels), struggle to learn how to write good specifications and identify mistakes made in their SRS documents. Our experience of teaching RE courses in the last 2 decades has provided many insights about the issues and challenges for REET. We have experimented with different pedagogical approaches to improve the learning outcomes for RE students. Our interest in conducting this study was not only inspired by our teaching experiences but also been triggered by the practical challenges of requirements inspection that has been communicated to us by our industry collaborators.

In this paper, we present the design and analysis of a pedagogical approach for teaching requirements inspection. We conducted an exploratory case study with 138 postgraduate students at University of Technology Sydney sorted in 34 teams to conduct requirements inspection collaboratively in an active-learning based pedagogy. We analyzed the data both quantitatively and qualitatively to assess the effects of the pedagogical design on students' learning. Our results show that there is a moderate statistical correlation between the performance in writing specifications and inspecting them. Furthermore, different challenging aspects were noticed by students when performing inspection such as managing inspection meeting time fruitfully, and the complexity of dealing with different levels of defects and with limited information. Important learning benefits have also been experienced by the students, such as teamwork, the importance of roles, and the relevance of systematic procedures as well as standardized forms and checklists.

This paper contributes a pedagogical design for teaching requirements inspection and its empirical evaluation. Students' reflections shows that not only they learnt to conduct requirements inspection but also conceded to have learnt from detecting the mistakes of others' through peer-review and self-reflection,

2332-6441/20/$31.00 ©2020 IEEE
DOI 10.1109/RE48521.2020.00032

which have been empirically demonstrated to be effective learning pedagogies in REET [6, 7]. Furthermore, the results emphasize the relevance of time, roles and engineered processes for requirements inspection.

The rest of the paper is organized as follows: Section II summarizes the background and related research work available on requirements inspection. Section III provides overview of the pedagogical design. Section IV gives details of the steps of research methodology and section V describes the results. Section VI discusses the implications of the research. Section VII provides limitations to the study and Section VIII conclusion and future directions.

II. Background and Related Work

For an inclusive discussion of the works in the field, in this section, we consider also papers addressing inspection of other software engineering artifacts (e.g., code, interviews), but the current work focuses only on requirements inspections

After Fagan [8] introduced the main stages and procedures to conduct successful design and code inspections, different approaches to detect faults through inspections were considered [9], and compared [10]. A survey on the topic was published by Arum et al. [11], in which different kinds of reading techniques, such as ad-hoc, checklist-based, defect-based, and perspective-based, were presented. Katasonov and Sakkinen [12] proposed a unified framework to analyze the quality of requirements and provided a characterization for reading techniques, adding scenario-based [13] and pattern-based approaches to those in [11]. More recent works focused on the challenges connected to requirements reviews in practice, that span from the long time required for their implementation [14], to the need to have more effective elicitation techniques [15]. This latter goal is pursued by Karras et al. [16], who developed a tool for video inspection of requirements workshops. In this direction, Ferrari et al. also proposed to analyze interview recordings [17].

Differently from the above-mentioned approaches, we are not proposing a novel inspection technique, but a new pedagogical approach to teach inspection to students. Focusing on teaching inspections is particularly relevant because, as shown in [18], the effectiveness of an inspection depends largely on the skills of the individual involved inspectors and their education. After conducting a large-scale controlled inspection experiment with over 70 professionals that focused on the relationship between an inspector's background and their effectiveness during a requirements inspection, Carver et al. [18] concluded that inspectors with university degrees in majors not related to computer science found significantly more defects than those with degrees in computer science majors. They also observed that the level of education (Masters, PhD), prior industrial experience or other job-related experiences did not significantly impact the effectiveness of an inspector. The only other type of experience that had a significant impact on effectiveness was experience in writing requirements. To overcome the general difficulties of teaching requirements engineering, Portugal et al. [19] presented a pedagogical approach for undergraduates, in which students during the semester can play different roles, including the auditor who is responsible to verify models through checklist inspection.

Not specifically focusing on requirements inspection, Schilling [20] discusses a novel active learning exercise to teach students how to perform and assess the effectiveness of software inspections. Students have to select an artifact from their capstone design projects and use fault injection to strategically place faults within the artifact. In this case, they do not have to perform the inspection themselves, but prepare an inspection packet consisting of a set of inspection instructions, applicable checklists, and the inspection artifact and "hire" their classmates, looking at their backgrounds and experiences. The team leader then holds two inspection meetings and reports the results. This exercise has been evaluated considering students feedback as very educational. Differently from our approach the focus is on "professional" choices, and what is missed in the reflection component.

In [21], Gazerani et al. present a framework to teach software inspections, composed of three parts: general guidelines, specific guidelines and learning activity. General guidelines include software inspection concepts and techniques. Specific guidelines include software inspection process and applying technique on it. A qualitative evaluation with the students at the University of Malaya shows positive impact on teaching software inspections.

The work closer to ours is [22], in which Goswami and Walia investigated the impact of reflections on inspection results to the students' understanding of the requirements inspection process and their abilities to find real software faults during the inspection. The authors measured the effectiveness of their approach having students working individually to inspect two documents seeded with defects, one before and one after the reflection, to measure the performance of the students. This approach and its evaluation differ from ours in many aspects, including the steps of the overall pedagogical approach, the size of the considered projects, the difference in kinds of problems in the SRS (natural versus seeded), the choice of group versus individual work, and the nature and structure of the required reflection.

III. Pedagogical Design

Several pedagogical approaches based on well-known learning theories in various combinations (depending on the educational context in SE), have been utilized by educators [23-31]. These learning theories can be traced back to the constructivist paradigm of educational philosophy [23]. Hence, 'Learning by Doing' [24], in combination with 'Situated Learning' [25], 'Discovery Learning' [26], 'Learning through Failure' [27, 32, 33] and 'Learning through Reflection' [7, 28] are the most widely used pedagogical design for SE curricula in order to meet the authentic assessment requirements of learning outcomes.

Authentic assessment requires the educators to design tasks for students while simulating the challenges of real-life work environments in which they have to focus on problem-solving skills based on their previously gained knowledge and the management practices [34, 35][36]. However, it is also the challenging part that SE and RE educators face, how to bring the right balance of 'realism' within the constraints of the academic environment [37, 38].

A. Pedagogical Context

In this study we present the pedagogical design and implementation for a module on requirements inspection as part of a postgraduate RE course at University of Technology Sydney. The assessments and tasks for this course are designed using the pedagogies of collaborative learning, authentic assessment [34, 39] role-playing [38, 40, 41] and contributing student pedagogy [42]. The overall assessment in this course involved three sets of tasks i.e. Requirements Elicitation, Requirements Specification and Requirements Inspection (See Fig 1).

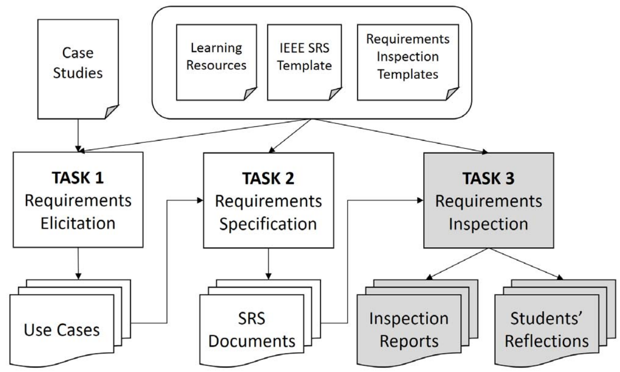

Fig. 1. Pedagogical Design

Prior to all three tasks, the students were grouped into teams of 3 (or 4) at the beginning of semester for the purpose of the collaborative learning for their assessment tasks: elicitation, developing use cases, and writing SRS. For requirements elicitation, students conducted a series of interviews with stakeholders (role played by teaching assistants), to understand the problem domain and discover the requirements. Then, they developed a set of use cases for the given case study. Once the use cases were developed and feedback received from the instructor, each team developed a complete SRS using the IEEE standard template.

B. Educational Task

In this paper we present the design and implementation of TASK 3. The complete set of educational resources can be found at [54]. The design is inspired by the seminal work of Fagan for code inspection [1, 43]. However, we are focusing only on requirements inspection. The learning for this module commenced by students attending a 2-hours' lecture about requirements validation in general, and formal inspection in particular. They also attended two (one-hour session) tutorials on requirements inspection where they were presented with examples of some types of defects followed by carrying out an exercise of finding defects in a very small SRS using checklists. The assessment task was designed to teach students the formal inspection meeting process and its related tasks as a method for requirements validation. The stages of this process were: planning, preparation, conducting the inspection meeting, and report writing. Each team was asked to conduct a formal inspection of an SRS that was developed by another team in the class for TASK 2 (see Figure 1). Groups were also asked to insert line numbers in the SRS they were supposed to inspect for ease of reference. The following instructions were given to students for the conduct of TASK 3:

- **Inspection method selection** - The lectures introduced different methods of organizing an inspection meeting (e.g. checklists, round robin, walkthrough, speed review). Students were asked to gather supporting documents, including checklists, role descriptions, defect recording forms, etc. While samples for most of these were made available online [24], students were told that they may search further to find others and then make their own selection.
- **Familiarization with the inspection forms** - Forms were available for documenting the defects discovered before and during the inspection meeting, and forms for summarizing the findings afterwards, this included an Issues Log and a Typo List from Karl Weiger's online resources [44].
- **Assign roles** - The roles include: leader or moderator (chairs the meeting), recorder or scribe (document the findings during the meeting), reader (reads the requirements out loud if needed), and inspectors (inspect the requirements).
- **Schedule the meeting:** Set a date, time, and place for the inspection meeting - Make sure all team members are available, allocate at least 2 hours uninterrupted time, and a quiet place to work.
- **Prepare for the inspection** – Each team member should prepare for the inspection meeting by carefully reading the SRS document before the meeting, and compiling an initial list of defects (typos and issues) found.
- **Conduct the inspection meeting** - If any team members are either absent or not prepared at the start of the meeting (based on the above), the meeting is to be postponed and an alternative date to be set. The inspection method (as in first point above) is used to carry out the inspection.
- **Summarize and record all the findings** – All the forms used in the inspection should be collected and defects discovered will be aggregated and summarized.
- **Write the inspection report** – This report describes the inspection process that was followed, key findings and defect lists (Typo list and issue log), reflection on the process and discussion of insights gained, both on the SRS, and the nature of the inspection process.

The submitted assignments were reviewed and marked by the Teaching Assistant using a pre-designed marking rubric that focused on the quality of the process and the validity of the output (i.e. defect list), from this formal inspection. Students were also asked to provide a *reflection report* for assessment.

IV. Research Design

In this section, we present the research design concerned with the evaluation of our pedagogical approach for teaching requirements inspection. We conducted an *exploratory* case study with 138 postgraduate students at <<*Anonymous University*>> grouped into 34 teams to conduct requirements inspection. This study aimed to answer the following research questions:

1. *What types of requirements issues and defects were found by the students?*

2. *Is there any correlation between the grades achieved for writing good quality SRS and the grades achieved for conducting good quality inspection task?*
3. *What are the benefits and challenges experienced by the students in requirements inspection?*
4. *What are the recommendations that can be derived from the student reports?*

The current list of questions is addressing the following overarching goal: "*Explore the emerging phenomena arising from the application of the proposed pedagogical approach for teaching requirements inspection*". Emerging phenomena from RQs are classified into

1. Descriptive aspects (RQ1), oriented to understanding the outcomes of the application of this approach,
2. Predictive aspects (RQ2), oriented to identify predictive patterns in the application of the approach, and
3. Improving aspects (RQ3 and 4), oriented to find information that is useful to improve the approach.

The data that we have analyzed consisted of:

1. *Inspection report submitted by students that included: description of their process, choice of inspection technique, the defect lists, and their overall reflections.*
2. *Marking rubric for SRS assignment (task 2).*
3. *Marking rubric for inspection (task 3).*

We analyzed the data both quantitatively and qualitatively to assess the effects of pedagogical design on students' learning. Students were informed that their assignment deliverables will be analyzed by the instructor for the purpose of research and evaluation of teaching method for the purpose of improvement. The dataset used in this study was de-identified and analyzed more than one year after the course was taught. The data collection had no impact on the grade of the students, this study was considered as negligible risk according to ethics guidelines.

For RQ1, we examine the results of the inspections from the typo lists and issue logs produced by the team. Then, we categorize the defects by type, and we compute the frequency of each type of defect.

For RQ2, we consider the grades obtained by the students' teams in the SRS writing activity and the grades obtained by the same teams in the inspection activity. Correlation analysis with Spearman's correlation coefficient is performed to evaluate the relationships between the two types of grades and understand whether teams whose SRS was of good quality are also good requirements inspectors.

For data associated to RQ3 and RQ4, we use the reflection reports of the different teams, in which they reflected on the lessons learned from the inspection experience. About two pages per team were produced, total of about 70 MS Word pages. To answer RQ3 and RQ4, we perform a thematic analysis of the reflection reports. We first split these reports into two random sets, and two of the authors separately performed thematic analysis through open coding on the sets. The analysis is oriented to identify themes associated with *challenges* and *benefits*, and to identify possible *recommendations* for students that could be directly derived from the reflection reports. The recommendations were also grouped into themes. The authors first worked individually on separate sets, and then cross-checked each other's work. They used an incremental approach and the codes were discussed and merged during the meetings between the coders until an agreement was reached. Finally, they homogenize the identified themes and come to a final set of challenges, benefits and recommendations. These themes were further reviewed by another author for consistency and coherence.

V. Results

RQ1. What types of requirements issues and defects were found by the students?

To teach students about requirements defects, they were given three sources of defect lists: "inspection issue log" from Wiegers [44] that gives 8 types of defects, University of Maryland's defect classes, and NASA Formal Inspection Guidebook. Students were also told that they can search and find other defect types from online sources to create their customised defect list. We then analysed submitted *issue logs* to discover how many groups just used the default list given on the Issue Log form and how many went further and incorporated defect types from other lists. The results are presented in Figure 2.

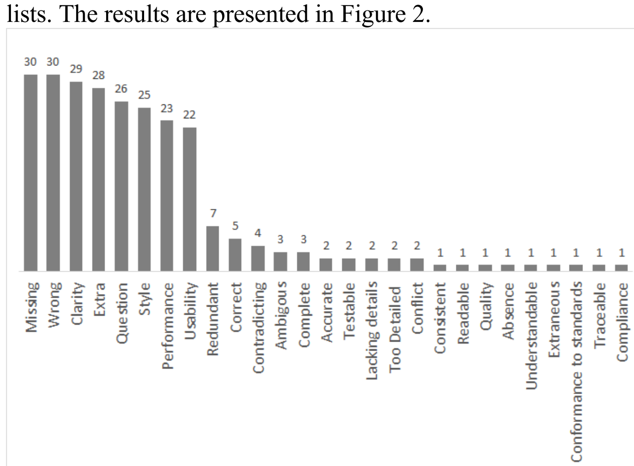

Fig. 2. Frequency of Requirements Defects types reported by students (N=34)

The terms used in Figure 2 are precisely those referred to in the inspection reports. However, some overlap is clearly observed, e.g. *completeness* and *lacking details* could mean the same defect, or *contradict* and *conflict* could be the same.

RQ2. Is there any correlation between the grades achieved for writing good quality SRS and the grades achieved for conducting good quality inspection task?

As shown in Figure 1, in our pedagogical approach, groups first develop and submit the SRS for their TASK 2 and then inspect the SRS of another group for their TASK 3. For TASK 2, students were given two different case studies, groups with even numbers worked on case study 1, and groups with odd numbers worked on case study 2. For the inspection task, each group worked on an SRS for the case study that they have not worked for TASK 2. After the completion of SRS and inspection tasks, students were evaluated through two well-structured grading rubrics by an experienced teaching assistant, who evaluated the work of all the groups for both tasks.

The marking rubric for TASK 2 includes 5 main sections: software requirements specifications, relevant models, traceability matrix, minutes of the stakeholder meeting, and quality of the quality of requirements elicitation interviews. The majority of the points are assigned in the quality of the specification (8 out of 13) that is the only part in the overall score that is relevant for our analysis. This part of the rubric is very detailed and covers the completeness and the quality of both the documents and the requirements themselves, so it highlights the quality of the SRS.

The evaluation rubric for TASK 3 includes 4 main sections: entry/exit criteria, description of the inspection process, result of the inspection exercise, and lessons learnt report. Since writing the lessons learnt report has pedagogical value, but it is not relevant to evaluate the quality of the executed inspection, we excluded it from this analysis (2 out of 10) from the evaluation rubric when answering our research question.

Hence, considering only the relevant components of the grades for TASK 2 and TASK 3, we performed the correlation analysis with Spearman's correlation coefficient (R_s) and we found that there is only a moderate positive correlation between the grades for good quality SRS and the grades obtained for good quality inspections ($R_s = +0.62$, p (2-tailed) = 0.00011).

However, there may be multiple confounding variables due to the real context, as this is a case study conducted in a real environment in which not all variables can be controlled.

RQ3. What are the benefits and challenges experienced by the students in requirements inspection?

As described in Sect. IV, two researchers analyzed the 34 documents of students' reflection on the inspection task in order to identify the themes concerning benefits for their learning, or challenges they faced that could be derived from their reported experiences. Similar to the three themes that emerged from systematic review of Walia and Carver [45] for taxonomy of requirements errors, we also found that the reflections were related to three main dimensions of the inspection task: **Process** (inspection activity), **Product** (the SRS under inspection) and **People** (the students involved in the activity). Figure 3 provides a summary of the thematic analysis from students' reflections by categorizing the benefits and challenges reported by the students into the above mentioned three categories along with their frequency of reporting from 34 groups.

In the following, we discuss the themes according to the identified dimensions, quoting relevant excerpts of the reports to exemplify the most relevant themes.

1) Process

a) Benefits

The benefits reported by students about the process of inspection were mainly related to their learning of the task by practically performing it in a realistic environment. Furthermore, students also reported to have understood what is expected from a good SRS based on the mistakes of others (peer assessment) or their own (self-assessment). Having the perspective of inspector of the SRS helped the students learn more about writing an SRS.

Although in the context of this study we did not assess whether the students actually became better SRS writers after the inspection experience, the following quotes (referred to the theme "Learning from mistakes of others"), highlight that at least they have reflected on each other's' mistakes:

"During the inspection, we [were] aware that we also made the same mistakes as others. For example, no[t] enough definition or description of acronyms and so on."

"We also can take their SRS document as a mirror to reflect our shortages."

"The exercise not only demonstrated the importance of inspections, but made the inspectors more aware of issues that their own requirements may have"

Some students made a step forward in their reflections and observed that they were able to take the viewpoint of the SRS reader (theme "Learned additional perspective of the reader of SRS"), and this change of perspective enabled them to learn about their own mistakes:

"*Through the process of inspecting other's work, we also learned to pay attention to our potential mistakes which might seem common but easy to neglect. We found that expression imprecision may be bothersome for readers, reviewers or inspectors alike. Thus, we took immediate action in our inspection final report and double check whether we have made our deliverable reader friendly and error free.*"

Other quotes highlight the relevance of the theme "Learning while doing", concerned with the benefits of the hands-on experience that pushes to take pragmatic choices based on the issues faced along the process:

"Looking back into the inspection process that was conducted as a part of assignment 3, has allowed us to gain a practical approach and provided some hands-on experience."

"The reader role was abandoned fairly early into the meeting, as it was found to be much more convenient for the person reciting their defects to read out the line they were referring to themselves, rather than delegating this job to a dedicated person."

"the checklist and walkthroughs were modified by the moderator, which helped focus on the defects rather than the unnecessary parts in the documents."

"[...] as our progress continued it has given us the right direction to find out the errors in the functional requirements"

Finally, the following quotes are concerned with the positive sentiment expressed by many teams regarding the overall learning experience ("Good learning experience"). These quotes highlight that the inspection task was not seen as a mere group exercise, but as a positive and engaging learning process:

"We all considered the meeting to be a success." … "On a personal level, we came out of the activity feeling positive."

"The thorough preparation also meant that downtime during the meeting itself was virtually eliminated, resulting in everyone being more engaged and active in discussion."

Freq (N=34)	Challenges	Category	Benefits	Freq (N=34)
18	Poor Time Management	Process	Learning form mistakes of others	7
15	Poor Planning		Good learning experience	7
12	Authors missing in the process		Learning by doing	4
4	Challenging task for novices		Mutual learning	2
4	working with limited information		Learned additional perspective of reader of SRS	2
1	Misunderstanding among team members		Active team planning and participation	1
1	Not full authenticity of real world		Learning from mistakes by self-assessment	1
6	Document format issues	Product	Improved the qualityof SRS	7
5	Too many typos		Increased learning on types of requirements defects	1
4	Difficulty in multi-level analysis		Learning how to write good SRS	1
4	limited information in SRS			
1	Difficulty in understanding SRS			
10	Poor Teamwork	People	Importance of roles	7
7	Workload with roles		Great teamwork	4
2	Miscommunication among team members		Great coordination	2
2	Conflict resolution with team members		Learning good communication	1
2	Futile discussions on issues wasting time			

Fig. 3. Percentages of the frequencies of the benefits and challenges reported by 34 groups classified against Process, Product and People

a) Challenges

The topmost challenge for students during this assignment was the time management and planning. These issues were raised in the context of collaboration among team members with different time schedules, most of them being novices for this task, limited information on SRS and not knowing how much the discussion for consensus would take. The absence of the authors of the SRS was pointed out to be one of the challenges as the students were not sure if their understanding of a defect was correct from the perspective of the authors of the SRS and the problem domain.

Concerning "Poor Time Management" and "Poor Planning" the following quotes illustrate how the students became fully aware of their problems with time during the different phases of the inspection process, and linked them to their weak planning:

"Even if we make a schedule, the actual execution time is more than we expected, we should increase our efficiency or make a more reasonable timetable."

"We find that we do not have sufficient time to do preparation. Because we did not realize the preparation is a big work which cost[s] much more time."

"The meeting session was challenging as there wasn't sufficient time to prepare, although each group member had the chance to read through the SRS before the inspection meeting, it was difficult to comprehend how long the entire process would take"

One of the main challenges encountered was also the "*Need to work with limited information*", due to communication problems, or related to issues of tacit knowledge [46-48] that every professional actor of the software process frequently experiences:

"We had dropped a group message to the group which we received the SRS from as we were expecting the meeting minutes of the third interview from them. But unfortunately, we haven't received it until the end. Hence, we continued with the meeting of minutes on only two stakeholder interviews."

"many requirements in the SRS did not relate to the case study and were not recorded in the answers of the stakeholder from the first two interviews."

"In our meeting when we were going through the different functional requirements of the SRS we had many questions which were raised in our minds because some of the requirements were not clear to us in the way that they were stated."

Several teams motivated these issues with the absence of the authors of the SRS documents during the inspection meeting ("Authors missing in the process"), which were seen as the source of the tacit knowledge or at least the means for clarifying unclear information:

"One of the limitations of the meeting is that we do not have the author present, which can be helpful in cases we need clarification on any ambiguities of the work product. The author can also explain their rationales during their writing so that we can provide feedback from our perspectives for faster and more efficient workaround or improvement"

"It's important for the Author of the SRS document to be present during the inspection meeting. This would have helped us immensely as we would have understood their perspective."

2) Product

a) Benefits

Concerning the SRS product itself, the main benefits are for the inspected product, as it was considered improved---or better, improvable based on the inspection---but also for the students themselves that better understood, from the SRS at hand, both the different types of requirements defects and the form of a good SRS.

Overall, the students felt confident that the inspection process has improved the quality of the SRS document under their review and considered it to be a very important activity ("Improvement of the SRS"), as shown by the following quotes:

"After the successful validation, our team was fully confident about the success of the quality improvement of the whole SRS as it was inspected very meticulously and precisely."

"According to the inspection, our teams feel that we can improve the quality of the work product significantly."

b) Challenges

Due to limited information in some SRS documents and various linguistic issues in writing, students had difficulty in understanding and carrying a multi-level analysis on the SRS documents that were written on various levels of abstractions.

The main challenges were associated to the presence of several typos and grammatical errors, which capture the attention of the inspector, may distract from the really relevant issues, and make the product appear unprofessional ("Too many typos"):

"Avoiding low-level errors like spelling errors and grammatical errors, or it will make people suspect quality of SRS"

"The truth is that we could not find a requirement is completely free from grammatical errors, because few classmates are native speakers, so that their requirements are hindered by language barriers, which is very troublesome to cross-check with the checklist"

"The typo form poses some negative effect on the inspection. I discourage that grammatical problems or style issues and those issues that [are] related to linguistics should [be] consider as a defect of requirements. However, it may pose a severe problem when developers are unable to understand what a specific requirement mean[s]"

Defects in the SRS document have different levels of abstraction and refer to different artifacts (natural language requirements vs diagrams), and the students experienced the challenge of the high cognitive load needed to analyze different kinds of mistakes, and the insufficiency of the existing tools ("Difficulty in multi-level analysis"):

"If the checklist is too specific, it would not be flexible enough to cover different circumstances. On the other hand, if the checklist is too general, it would become too vague to clearly describe the defects. Thus, we need to keep the balance between these two scenarios and also integrate other methods such as Ad Hoc and Walkthrough to enhance the productivity in our process."

"Since the main formats for issue documentation were the Issue log and the Typo List, there were two categories of issues to go through. Due to the high amount of identified issues, focusing on one of the two formats in a single session could have been more beneficial."

Some of the groups also highlighted the need to deal with format issues. The SRS documents received were in PDF and it was difficult to identify the lines to point to corrections, especially in case of tables or diagrams ("Document format issues"):

"Without inserting line numbers to a selected SRS document, then you would not be able to locate a typo or an issue accurately"

"Another time-consuming difficulty was represented by putting line number into the pdf document."

"[...] inserting line numbers makes things even harder, as tables are unable to be numbered or ordered, which is hard to name an error specifically [...] Hence, after converting the received pdf document with the help of a free tool, some parts (especially the use case tables) were printed incorrectly and offset. These layout problems caused by the tool had to be resolved manually which took a lot of time and effort"

"Word document with line numbers was then re-formatted to pdf to ensure all group members had the exact same document without the risk of making unwanted alterations to it."

3) *People*

a) Benefits

Some groups expressed satisfaction about their teamwork and coordination ("Great Teamwork", "Great Coordination"), while others emphasized how the presence of different, well-prepared, roles were one of their main strengths as a team ("Importance of Roles"). The following quotes document the importance of the people dimension in the experience:

"Although this meeting kept more than five hours, we not only analyzed article of Group 30 but also learned how to be better in cooperation. Luckily, each group member not only completes the personal tasks, but also helps others to improve our inspection report." ("Great Teamwork")

"Since the inspection meeting had a moderator it led to a central authority which could establish rules and communicate with the group more effectively. This leads to less digression from the topics." ("Importance of Roles")

"The different roles assigned to each member of the team helped recreate an industry scenario where we are expected to collaborate and work in teams." ("Importance of Roles", "Great Coordination")

"The dedicated recorder contributed to maintaining the meeting's momentum, as it wasn't necessary for people to constantly stop and write things down on their computer during discussions." ("Importance of Roles")

b) Challenges

Although some teams were happy with the collaborative teamwork and communication and how they learned to organize an inspection activity, a lot of groups reported that they struggled with the teamwork ("Poor teamwork") specifically with the aspect of defining clear roles with clear duties ("Different roles have different workload"). A lot of their time was wasted due to miscommunication among team members or futile discussions during the conflict resolution ("Conflict resolution among team members", "Miscommunications among team members") hence suggesting the importance of a meeting moderator among the team members to keep everyone on track for agenda of inspection meeting. The following quotes illustrates the typical challenges faced by the students at the level of interpersonal relationships and role management:

"A teamwork significantly becomes to play a major role during executing this activity. Our group was confronted with the time consistency to the meeting, some different attitude or perspective during discussion and assigned tasks issue" ("Poor Teamwork")

"Working in a group can be problematic when the group members are not co-operative but in our case all the members were co- operative. The problem was due to difference in our timetable" ("Poor Teamwork")

"we found the work assigned to each member had different workload which lead to some delay happening in the whole assignment process." ("Different roles have different workload")

"some roles require more pre-work before the meeting, some roles are less pre-work but under high pressure of after work." ("Different roles have different workload")

RQ4. What are the recommendations that can be derived from the student reports?

From the themes identified and associated quotes we can also derive some recommendations that may be useful for other students that have to deal with requirements inspection. Again, we divide the recommendations into the three dimensions of process, product and people.

1. Process

The students found particularly useful, and therefore recommended, the use of structured tools for the process, especially standardized forms and checklists for the inspection. The already mentioned presence of authors and the shared understanding of the problem domain and SRS document were also considered important for the task.

The adoption of checklists, standardized forms, template and guidelines were considered as the major tools to guide the process and ensure productivity, and also to let all members have their voice (see last quote):

"These forms are more useful than we thought before, especially the Issue forms. It details where the issue comes from…, and identifies what type it is …, and highlights the level of severity …. This significantly serves as a guidance on what an inspection should look into..."

"All members used the checklist for the work product before meeting, and we found it was really helpful for organizing inspection structure and carrying the work"

"One document that was quite useful in finding the errors was SRS template as it has a brief description of what each section of the document is about."

"Before the inspection meeting, each team member went through the SRS Document rigorously and penned down problems in a specific format provided by our Moderator before the inspection meeting. Adopting to a single format made our inspection meeting very economical in terms of time and productive."

"in fact, we felt that some of the quieter team members were able to contribute more than they otherwise would have thanks to this format."

Other recommendations are concerned with the need for a more uniform preparation to improve shared understanding among team members:

"It would have made the whole inspection process a bit easier if everyone had had a solid understanding of the whole process and roles of inspection procedure before meeting for the first time"

"it would be worth it to spend the time clarifying what each of the [defect] classes exactly means, so that everyone in the group is on the same page."

Another recommendation coming from the experience is related to the need to have an appropriate individual preparation on the SRS before coming to the inspection meeting:

"Each inspector prepared at least three hours before [the] inspection meeting, so we could raise defects and issues efficiently"

"Everyone should come half an hour before the scheduled meeting time and then have to read their parts from the SRS document to be inspected, this not only helped to get the idea of the case study on which the SRS has written but it also saved our time in total."

"We also had to allocate additional time to re-read the SRS to make sure the inspection we were doing was on the right track and it would have been more successful if we had prepared for a few hours each beforehand or re-read the SRS a couple of times."

2. Product

The students reflected on their experience to suggest that the knowledge of the problem domain for which the SRS is written could resolve some of their challenges they faced during inspection. The students also suggested to be presented with examples of high-quality SRS to have an idea of the benchmarking that they have to raise their inspected SRS documents to.

"The members should spend almost one day to read SRS document before and record the errors and fill them in the typo list before the meeting"

"the document that came was not with line numbers, and spell checked. Entering line numbers as well as running a spell check on the initial document can save a lot of time for the inspection team, as they can focus on the other more important areas."

"a sample of high-quality SRS provided before the meeting can be helpful because we can refer to it as a guideline of expectation for this type of work product

3. People

Students reflected that they could have benefited from using communication tools for team collaboration more effectively and hence would have had better planning and time management for the activity. Furthermore, they also noticed the relevance of other remote communication tools (Skype, Whatsapp, etc.), not considered in their guidelines for the:

"Even after the inspection meetings our group were communicating with each other with the help of phone calls and a private WhatsApp group."

"Any task that was left unfinished was allocated by the moderator and was communicated though Facebook Messenger"

"Also, our team made use of associate communication methodology which incorporated on- line chatting with team members, visual meetings, and phone discussion"

"After that place should be changed to some collaborative team study area where there is availability of some screen projection so that one of the group member can connect laptop and every one can easily see the inspection report"

Other recommendations that can be extracted from the report are related to solutions to the previously listed challenges: the need to define clear roles to improve efficiency ("*Defining clear roles will reduce effort and time*"), and the paramount importance of teamwork ("*Good teamwork critical to the task*"):

"A good collaboration between the inspection members and their attendance, is highly important for the success of the formal inspection process." … "We think the skills of communication and teamwork are more crucial than the assignment."

VI. Discussion

The *main take-away messages* of our study, in relation to the RQs, are:

1) Most of the defects found in the SRS concern Missing, Wrong and Unclear requirements. Incomplete or hidden requirements (which can be regarded as "Missing") are also one of the most common issues identified by the NaPIRE survey [50]. Furthermore, Communication problems (which can be associated to Unclear requirements), are one of the most common problems in RE, according to the same study. So, apparently, novices and experienced requirements engineers suffer from similar problems, regardless of the fact that the responsibility resides on the requirements editor or on the inspector (who may not have sufficient domain knowledge to clarify and filling the blanks). The frequency of Wrong requirements, instead, may be associated to the need of students to have a simple label to identify generically defective items.

2) A moderately positive relationship occurs between the ability of a team to write an SRS and the ability of the team to perform a high-quality inspection. This is not surprising, as the two aspects have been seen to correlate also in industrial practice as shown by Carver et al. [18]. Although other confounding variables may have impacted on this result, it is interesting to see that similar patterns occur between novices at their first experience and professionals.

3) Most of the challenges identified by students are related to time and planning. This is not surprising, as the time to dedicate to RE activities is also one of the main problems identified by professionals [50]. This shows that the proposed pedagogical approach enables students to experience the frustrations of real requirements analysis. They also frequently experienced the difficulty of working in teams, but also learned teamwork throughout the study. Among the other benefits, the student particularly enjoyed the possibility of learning from the mistakes of others, and articulated self-gratification for being able to improve the quality of the SRS.

4) The recommendations for other students are mainly concerned with the need to take outmost care of the preparation: reading the requirements before the inspection meeting, having line numbers in the PDF files, and make the most of the checklist and structured forms made available. Overall, to deal with the challenges with time, students understand that the inspection process needs to be carefully engineered.

Below, we also discuss *improvement of the proposed teaching method*, based on existing studies.

Over the years, software engineering education researchers have proposed alternative approaches to industry-based learning, by designing curriculum and task activities based on project-based learning and authentic assessment principles [34, 39, 51]. These approaches stress the need for the design of activities to be based on 'realistic' problems that students have to solve in a collaborative environment, thus simulating the real-world environment within the classroom. There is still the challenge of how far the educators can go in providing 'authenticity' in their curriculum and tasks [51], considering that the students are not yet fully trained to embrace all the challenges of the real world. To prepare the students for the real working environment, Dawson [37] has proposed a list of 20 tricks in order to help the educators determine the level of authenticity in their assessments and tasks in project-based learning (e.g. conflicting requirements, hardware crash, or uncertain customers). However, there is a varied level of authenticity that can be achieved in different software engineering courses depending on the types of assessments, resources and level of students [52].

We observe that 4 out of 20 tricks of authenticity from Dawson's list can be mapped to our Inspection TASK as shown in TABLE I. These tasks are further mapped to the skills that students required for this assessment task as sown in TABLE II. From the two tables we can see that in our inspection task students were challenged for their *problem-solving skills, people handling skills, negotiation skills, planning skills, adaptability, quality understanding, organizational skills and design skills.*

TABLE I. Comparing Authenticity of our Inspection TASK with Dawson's [37] Recommendations

Dawson [22]	Dawson's [22] Justification	Our Inspection Task	How in our Inspection Task
Give an Inadequate Specification	In the real world complete, unambiguous specifications are rare if they ever exist	SRS provided for inspections were written by students in previous task.	The SRS documents produced by students were not perfect and all the groups had to work with limited information
Present Customers with Conflicting Ideas	This type of conflict shows that more than one viewpoint in a problem is possible, and that developers need to communicate with all users and stakeholders to build a complete picture of requirements.	Inspection meeting	The absence of the SRS author in the meeting made it clear that they lacked stakeholders and information and had to do role-playing which was not providing them with a complete picture
Introduce Quality Inspections	Many students pride themselves in being able to produce "high quality" software. In reality code comments and documentation are often produced at the end of the project despite what they may claim.	Inspection process	Students had to research and develop an inspection technique to adopt for the inspection process. By looking at each other mistakes they admit to have discovered more mistakes than expected.
Change the Working Procedures	Changes in management personnel and procedures are not experienced at university where changes in the teaching staff or teaching conditions would not normally occur during the course of a student project.	Multiple guidelines for inspection and need to define roles and inspection process left open	Leave choice to change the roles and the checklists types.

TABLE II. Mapping Skills to Inspection Task Authenticity

Dawson's Tricks	Problem Understanding	People Handling Skills	Negotiation Skills	Planning Skills	Adaptability	Quality Understanding	Organizational Skills	Design Skills
Give an Inadequate Specification	X	X	X				X	
Change the Requirements and Priorities	X		X	X	X			X
Introduce Quality Inspections				X		X	X	
Change the Working Procedures		X			X		X	

Based on the analysis of our study and using the lens of Dawson's tricks that are not covered in the current pedagogical design, we offer the following lessons learned as instructors, which can be useful to adapt the course module:

- Modify the activity to involve one of the authors of the SRS to attend the meeting. This could be done after the inspection has taken place, to clarify missing aspects and possibly correct the SRS, otherwise students will not experience the realistic need to work with limited information. Presence of the author of SRS at the inspection meeting may increase the time of the inspection meeting, especially if the inspectors keep referring to the author to clarify or confirm.
- Planning and Time management are critical in this task for students to perform the inspection. Hence, instructors should emphasize the need to do proper planning, and, at the same time, change schedules and context, as it happens in real environments as suggested by Dawson's work.
- Knowledge of the SRS problem domain prior to inspection would help. One team member can be assigned the role of domain expert or product owner, and acquire knowledge of the domain beforehand to be able to provide answers to domain specific questions in the inspection meeting. Systematic walkthroughs of simple scenarios with limited domain knowledge [53] may be more effective in this scenario rather than checklist technique [49].
- Use of communication tools can improve the collaborative work. Therefore, the usage of collaborative platforms should be encouraged.
- Defining clear roles and having moderator at the meeting are important for collaborative teamwork to succeed. The use of roles should therefore be encouraged, but all subjects should also have the possibility of playing different roles, so that issues related to imbalanced work do not emerge.
- Use of standard forms and checklist and shared understanding among team members can improve the outcome of the inspection task for students. Instructors should put high emphasis on the need for standardization, but should also leave the students free to build their own standard process.

VII. Threats to Validity

A. Construct and Internal Validity

We conducted the study under interpretive paradigm where the findings are subjective to the context of the classroom and the understandings of the instructors and the researchers. However, we have provided full details of the pedagogical and research design and have tried to mitigate the subjectivity wherever possible in the research design. For RQ2, confounding factors may have affected the identified correlation. This threat could not entirely be mitigated, but it is justified by the exploratory nature of the case study.

B. Reliability

To improve reliability, two of the researchers qualitatively analyzed students' reflections and discussed to resolve differences on themes emerging from the data, thus increasing the reliability of the findings thanks to triangulation. Marks were assigned to students by a Teaching Assistant, who is not part of the research team, this eliminates the bias for the analysis of RQ3.

C. External Validity

The study was conducted in a multicultural classroom environment with a large number of international students with English as their second language. This may impact the results for generalization, however, considering the current SE industrial outlook with globalization and outsourcing where multicultural and diverse teams are working on software projects, this was yet another authentic element of the exercise for the students to learn and work with people from heterogeneous backgrounds.

VIII. Conclusion and Future Direction

This paper presents a pedagogical design for teaching requirements inspection to novices, together with an evaluation of its learning effect on the students according to their feedback. The study shows that novices have reported on learning and improving several important aspects of the inspection task, including the importance of roles, the need for strict time management, the relevance of standardized forms and checklists for guidance and the need to work with limited and ambiguous information. Furthermore, the study also shows that good inspectors tend also to be good SRS writers. Our future work includes further improvements in the pedagogical design, with the addition of more realistic aspects according to Dawson's tricks [37], such as changes of schedule and roles. Furthermore, we aim to study in more detail which are the typical mistakes made by students in SRS that led to generation of requirements defects, and which classes of defects are easier or harder to identify for novices. Finally, we have considered introducing another SRS writing exercise, albeit smaller than TASK 2, to investigate whether or not students SRS writing quality improves after having conducted a formal requirements inspection according to our pedagogical design.

REFERENCES

1. Fagan, M.E., *Design and code inspections to reduce errors in program development.* IBM Systems Journal, 1999. **38**(2.3): p. 258-287.
2. Boehm, B. and V.R. Basili, *Top 10 list [software development].* Computer, 2001. **34**(1): p. 135-137.
3. Anu, V.K., et al. *Effectiveness of Human Error Taxonomy during Requirements Inspection: An Empirical Investigation*. in *SEKE*. 2016.
4. Laitenberger, O., *A survey of software inspection technologies*, in *Handbook of Software Engineering and Knowledge Engineering: Volume II: Emerging Technologies*. 2002, World Scientific. p. 517-555.
5. Ouhbi, S., et al., *Requirements engineering education: a systematic mapping study.* Requirements Engineering, 2015. **20**(2): p. 119-138.
6. Bano, M., et al., *Teaching requirements elicitation interviews: an empirical study of learning from mistakes.* Requirements Engineering, 2019. **24**(3): p. 259-289.
7. Ferrari, A., et al. *Learning Requirements Elicitation Interviews with Role-Playing, Self-Assessment and Peer-Review*. in *2019 IEEE 27th International Requirements Engineering Conference (RE)*. 2019. IEEE.
8. Fagan, M., *Design and code inspections to reduce errors in program development*, in *Software pioneers*. 2002, Springer. p. 575-607.
9. Shull, F., I. Rus, and V. Basili, *How perspective-based reading can improve requirements inspections.* Computer, 2000. **33**(7): p. 73-79.
10. Porter, A.A., L.G. Votta, and V.R. Basili, *Comparing detection methods for software requirements inspections: A replicated experiment.* IEEE Transactions on software Engineering, 1995. **21**(6): p. 563-575.
11. Aurum, A., H. Petersson, and C. Wohlin, *State‐of‐the‐art: software inspections after 25 years.* Software Testing, Verification and Reliability, 2002. **12**(3): p. 133-154.
12. Katasonov, A. and M. Sakkinen, *Requirements quality control: a unifying framework.* Requirements Engineering, 2006. **11**(1): p. 42-57.
13. do Prado Leite, J.C.S., et al., *Scenario inspections.* Requirements Engineering, 2005. **10**(1): p. 1-21.
14. Salger, F. *Requirements reviews revisited: Residual challenges and open research questions*. in *2013 21st IEEE International Requirements Engineering Conference (RE)*. 2013. IEEE.
15. Femmer, H., et al. *Quality assurance of requirements artifacts in practice: A case study and a process proposal*. in *International Conference on Product-Focused Software Process Improvement*. 2016. Springer.
16. Karras, O., S. Kiesling, and K. Schneider. *Supporting requirements elicitation by tool-supported video analysis*. in *2016 IEEE 24th International Requirements Engineering Conference (RE)*. 2016. IEEE.
17. Ferrari, A., et al. *Interview review: detecting latent ambiguities to improve the requirements elicitation process*. in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. 2017. IEEE.
18. Carver, J.C., N. Nagappan, and A. Page, *The impact of educational background on the effectiveness of requirements inspections: An empirical study.* IEEE Transactions on Software Engineering, 2008. **34**(6): p. 800-812.
19. Portugal, R.L.Q., et al. *Facing the challenges of teaching requirements engineering*. in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. 2016. IEEE.
20. Schilling, W.W. *Teaching software inspection effectiveness: An active learning exercise*. in *2012 Frontiers in Education Conference Proceedings*. 2012. IEEE.
21. Gazerani, N., et al. *Developing a Teaching Framework to support software inspection*. in *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. 2015. IEEE.
22. Goswami, A. and G.S. Walia, *Teaching Software Requirements Inspections to Software Engineering Students through Practical Training and Reflection.* The ASEE Computers in Education (CoED) Journal, 2016. **7**(4): p. 2.
23. Navarro, E.O., *On the role of learning theories in furthering software engineering education*, in *Instructional design: Concepts, methodologies, tools and applications*. 2011, IGI Global. p. 1645-1666.
24. Dewey, J., *Education and democracy*. 1916, New York: Macmillan.
25. Lave, J., *Cognition in practice: Mind, mathematics and culture in everyday life*. 1988: Cambridge University Press.
26. Bruner, J.S., *On knowing: Essays for the left hand*. 1979: Harvard University Press.
27. Schank, R., *Virtual Learning. A Revolutionary Approach to Building a Highly Skilled Workforce*. 1997: ERIC.
28. Schon, D., *Educating the beginning practitioner*. 1987, San Francisco, CA: Jossey-Bass.
29. Moore, M. and C. Potts. *Learning by doing: Goals and experiences of two software engineering project courses*. in *Conference on Software Engineering Education*. 1994. Springer.
30. Tvedt, J.D., R. Tesoriero, and K.A. Gary. *The software factory: combining undergraduate computer science and software engineering education*. in *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*. 2001. IEEE.
31. Germain, T., P.N. Robillard, and M. Dulipovici. *Process activities in a project based course in software engineering*. in *32nd Annual Frontiers in Education*. 2002. IEEE.
32. Bano, M., et al. *Learning from mistakes: An empirical study of elicitation interviews performed by novices*. in *2018 IEEE 26th International Requirements Engineering Conference (RE)*. 2018. IEEE.
33. Donati, B., et al. *Common mistakes of student analysts in requirements elicitation interviews*. in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2017. Springer.
34. dos Santos, S.C. and F.S. Soares. *Authentic assessment in Software Engineering education based on PBL principles a case study in the telecom market*. in *2013 35th International Conference on Software Engineering (ICSE)*. 2013. IEEE.
35. Herrington, J. and A. Herrington, *Authentic assessment and multimedia: How university students respond to a model of authentic assessment.* Higher Education Research & Development, 1998. **17**(3): p. 305-322.
36. Gulikers, J.T., T.J. Bastiaens, and P.A. Kirschner, *A five-dimensional framework for authentic assessment.* Educational technology research and development, 2004. **52**(3): p. 67.
37. Dawson, R. *Twenty dirty tricks to train software engineers*. in *Proceedings of the 22nd international conference on Software engineering*. 2000.

38. Yusop, N., Z. Mehboob, and D. Zowghi, *The role of conducting stakeholder meetings in requirements engineering training.* Proc. of REET, 2007. **7**: p. 48-55.

39. Tai, G.X.-L. and M.C. Yuen. *Authentic assessment strategies in problem based learning*. in *Proceedings of ASCILITE*. 2007. Citeseer.

40. Zowghi, D. and S. Paryani. *Teaching requirements engineering through role playing: Lessons learnt*. in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.* 2003. IEEE.

41. Svensson, R.B. and B. Regnell, *Is role playing in Requirements Engineering Education increasing learning outcome?* Requirements Engineering, 2017. **22**(4): p. 475-489.

42. Herman, G.L., *Designing contributing student pedagogies to promote students' intrinsic motivation to learn.* Computer Science Education, 2012. **22**(4): p. 369-388.

43. Fagan, M., *Design and code inspections to reduce errors in program development.* IBM Systems Journal, 1976. **15**(3): p. 182-211.

44. *https://processimpact.com/goodies.html*. 19 February 2020].

45. Walia, G.S. and J.C. Carver, *A systematic literature review to identify and classify software requirement errors.* Information and Software Technology, 2009. **51**(7): p. 1087-1109.

46. Ferrari, A., P. Spoletini, and S. Gnesi, *Ambiguity and tacit knowledge in requirements elicitation interviews.* Requirements Engineering, 2016. **21**(3): p. 333-355.

47. Grunbacher, P. and R.O. Briggs. *Surfacing tacit knowledge in requirements negotiation: experiences using EasyWinWin*. in *Proceedings of the 34th annual Hawaii international conference on system sciences*. 2001. IEEE.

48. Gervasi, V., et al., *Unpacking tacit knowledge for requirements engineering*, in *Managing requirements knowledge*. 2013, Springer. p. 23-47.

49. Fusaro, P., F. Lanubile, and G. Visaggio, *A replicated experiment to assess requirements inspection techniques.* Empirical Software Engineering, 1997. **2**(1): p. 39-57.

50. Wagner, S., et al., *Status quo in requirements engineering: A theory and a global family of surveys.* ACM Transactions on Software Engineering and Methodology (TOSEM), 2019. **28**(2): p. 1-48.

51. Savery, J.R. and T.M. Duffy, *Problem based learning: An instructional model and its constructivist framework.* Educational technology, 1995. **35**(5): p. 31-38.

52. Abad, Z.S.H., M. Bano, and D. Zowghi. *How much authenticity can be achieved in software engineering project based courses?* in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. 2019. IEEE.

53. Mavin, A. and N. Maiden. *Determining socio-technical systems requirements: Experiences with generating and walking through scenarios*. in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.* 2003. IEEE.

54. Bano, M., Zowghi, D., Ferrari, A., Spoletini, P. (2020, May). *Inspectors Academy: Pedagogical Design for Requirements Inspection Training* - Educational Material. Zenodo. http://doi.org/10.5281/zenodo.3842861