Validation of Software Requirements Specifications by Students

Alberto Sampaio(✉) and Isabel Braga Sampaio

Interdisciplinary Studies Research Center (ISRC),
Institute of Engineering of Porto – Polytechnic of Porto (ISEP/IPP), Porto, Portugal
{acs,ais}@isep.ipp.pt

Abstract. Context: Requirements validation is an important activity in software development, but frequently neglected in computing courses. This paper describes a study where requirements specifications produced by students are validated by their peers.

Objective: To evaluate how well students perform the validation of requirements specifications.

Methods: A correlational study.

Results: Group' grades correlate with teachers' marks for a level of significance of 5%.

Conclusion: Students reviews can be used for peer assessment, but with some care.

Keywords: Requirements · Requirements specification · Validation · Peer review · Inspection

1 Introduction

Software requirements engineering is a basilar activity for software development [1] and its quality considered important [2]. Boehm [3] showed that the cost of defect repairing increases significantly along the software product life cycle, which implies that errors in requirements should be detected as early as possible. Concomitantly, cost factors require the implementation of the just needed requirements demanded by the client [4]. In an iterative development model, the software engineer is concerned with smaller sets of requirements, usually a subset by iteration. Smaller sets makes requirements validation lighter for each iteration, although a validation for a larger initial set (e.g. backlog) can be needed. Even if agile methods use distinct ways to validate requirements (e.g. acceptance criteria) validation is still performed and needed as shown in a meta study [2].

In a more traditional development model the engineer must do an initial validation of all elicited requirements. Traditional models like waterfall have not being abandoned. Recently there was an increase in the use of hybrid methodologies [5]. The emphasis on requirements elicitation at upfront is also a characteristic of model based development. To sum up, independently of the development

H. Florez and S. Misra (Eds.): ICAI 2020, CCIS 1277, pp. 505–518, 2020.
https://doi.org/10.1007/978-3-030-61702-8_35

model, requirements validation needs to be performed reinforced by the recent increase use of hybrid methodologies.

Following the importance of requirements validation and its possible benefit for the classroom (see e.g. [6]) it was decided to introduce requirements validation in a course. It was also decided to introduce it through a practical exercise of requirements review. The review has been considered an adequate method for requirements validation (e.g. [7]). Reviews can be more or less formal, being the inspection the more formal. There are several kinds of inspections and in this initiative it was decided to opt for one based on a checklist, but without all the formalism of a common inspection. Furthermore, the checklist is one of the methods also used in agile projects [2]. Another important motivation for this research is that even if students can badly design, they are considered able to evaluate others designs [8]. Being able to evaluate other designs, we extrapolate that students should also be able to evaluate others specifications. All this is reinforced by the fact that peer assessment is considered an effective way of assessment [9]. The main goal of this research is to know how well students perform requirements validation. To that, it was performed an empirical study where students validate requirements specified by their peers. The study is the main subject of this paper.

The paper is organized as follows. After this introduction, background and related work are reviewed in Sect. 2. Section 3 presents the empirical study. The paper finishes with the main conclusions in Sect. 4.

2 Background and Related Work

Review is a general term referring to defect detection by the examination of software work products (artefacts), including requirements. The term defect is used in a broad sense that includes quality criteria the requirements specification should satisfy. The most common types of reviews are inspections, walkthroughs and actual reviews [10] by descent order of formalism. An inspection is a kind of review but with a well defined process where participants follow specific roles. Inspections can be ad-hoc, checklist based reading or scenario based reading [11] according the technique employed.

In the case of ad-hoc inspections there is not a systematic way of inspecting the artefacts. In the case of a checklist based technique the inspectors follow a list of certain characteristics or classes of defects to search for. For a requirements specification a checklist would a list of items about the document specification the reviewer should focus, like correctness, priorities or others. Scenario based means that participants follow specific scenarios of possible defects according their expertise. Each scenario provides more detailed guidance to the inspectors than a checklist. A common scenario based technique is perspective based reading (PBR) [12] where the artefacts are read from the perspectives of the roles of the software stakeholders, like user and tester.

The use of reviews to teach requirements validation is not original. For example, in [6] the authors used a N-fold inspection with that aim. In a N-fold inspection the requirements document is reviewed by several groups and their results compared in a meeting. In [6] each group of students could choose between two techniques: checklist and PBR. After the exercise the students answered a survey and, from the survey results, the authors considered that the exercise goals were achieved and that the exercise did contribute to students have a better understanding of the engineering requirements process.

It is also common to use students as experimental subjects in requirements validation studies. An example is [10] where the authors performed 2 experiments with students to determine if checklist based inspections would improve defect detection of use case models. In the first study the use of a checklist was considered beneficial. In the second, their authors concluded that a checklist (p. 133) "may not be particularly useful when inspectors already have good knowledge of the defects they are expected to find as had the inspectors in this case". Also in the second experiment, the students with the checklist found more defects than the students that did not use it. The used checklist was based on a taxonomy of defects developed by the authors and is specific for use case models.

There are several tools available for requirements engineering (e.g. DOORS) but these tools, although providing help, do not allow the full automation of the process. This is also true for requirements reviews. In the case of agile approaches, there are tools (e.g. Cucumber) that can be useful to express requirements and to validate their implementation but not very helpful to evaluate the quality of the specifications. Recent research intend to develop new tools as discussed in [13] that can evaluate some requirements characteristics. Automation is essential in modern code reviews (e.g. [14] and so more automation is needed to achieve what could be called "modern requirements reviews". Anyway, requirements validation still remains a process requiring significant human intervention.

In the present paper, besides the use of a review to teach requirements validation, the review is performed by students' peers.

3 The Study

3.1 Background and Methodology

The main goal of this study is to find how well students perform requirements validation, more specifically, of software requirements specifications (SRS) documents.

The study was performed in a course from the third semester of an informatics program. This is a special course because it is implemented as a project with a duration of approximately 4 weeks including the final evaluation. Each week is an iteration. Before the project start, the students are grouped in teams. During the project one of the team members speaks directly with the teacher responsible for the course who acts as the client. The meetings with the client are face-to-face and occur at least once by week. The meetings involve all the groups in simultaneous, so all have to solve the same problem.

In the project, most of the requirements have been elicited at the first week, but new requirements have been added weekly. During the course, students produce several artefacts, in particular use cases to model software requirements.

The result of their work during the first week (iteration) is a document with the software requirements specification (SRS). The study focus is on the SRS document and not on the use case descriptions alone. After the students submit the SRS, the SRS was marked by the teachers and supplied a draw of the expected solution.

The initiative of introducing requirements validation in the course was implemented via inspections performed by students using a checklist. In the exercise the students only review the specification, that is, they not perform any other inspection role. Because students are organized in teams, the teams (or groups) are the experimental units.

This study was elaborated as a case study [15] in a learning context. As such, the study followed several steps, namely, preparation, data collection, analysis of the data and reporting. This research has an exploratory intent and, as such, the study can be considered as exploratory.

3.2 Research Questions

To validate the success of the initiative, it is necessary to evaluate the quality of the reviews performed by the students. If reliable, these reviews can be used as a form of peer assessment.

RQ1 - Do students perform well requirements reviews?

The answer to that question can also give some light to the viability to classify students works with the help of the classifications produced by their own peers.

Besides knowing if the reviews performed by the students can be considered of quality, it was decided to have a deeper understanding of the phenomenon. That is, to know if students "quality" affects the quality of their validation work. This second goal was considered relevant in order to find some indicator of trust that can be assigned to the results of students validations, so these results can be used for peer assessment. The goal was translated into a second research question.

RQ2 - There is a relation between the quality of the group and the quality of its validation?

Because there was no access to the marks of the students in other courses of the program, the quality of the groups is restricted to the course so far and specifically to their SRS documents.

3.3 Environment and Participants

The participants are all the groups formed at the beginning of the course. Context factors should be considered as they can play an important role [16]. At the time of the study, the knowledge area of requirements engineering was part of the program syllabus and taught in the previous semester, the second one. At the beginning of the study, students had no experience with requirements validation. Because all students took the same previous course, the experience of the students in the use of UML, particularly for requirements analysis, and their academic experience using the methods should be similar. However, it was known that a small number of participants had industrial experience in software development, but this was not taken into consideration for the study.

For the course, it was a defined a template for the SRS document to be produced by the teams. Not all teams completely followed the template.

Briefly, the main topics of the template are:

- System objective;
- Glossary;
- Use case diagram;
- Table with requirements and their priorities and possible dates;
- Actors/use cases;
- Constraints;
- Conceptual diagram (domain model).

Students could develop their specifications using any tool they know. For the study we were not concerned with tools usage and no tool was used to accomplish the reviews. The study was conducted during the normal classes schedule, so for teachers it was a regular class. The course involves fifteen teachers.

3.4 The Instrument

Students are from the 3rd semester and being in the first half of the course, they are expected to have a limited experience about software development and, as pointed before, they had no experience in requirements validation. A checklist seemed to be adequate because it provides concrete guidance [6]. Furthermore, the use of well designed check-lists in learning is a recommended approach to support students learning and performance [17] as intended with this exercise.

The quality of a requirements specification possess several dimensions and so it was necessary for the developed checklist to covers such dimensions. According the IEEE 830 standard [18] the characteristics of a good SRS are:

- Correct;
- Unambiguous;
- Complete;
- Consistent;
- Ranked for importance and/or stability;
- Verifiable;
- Modifiable;
- Traceable.

Those characteristics plus some guidance about quality issues with requirements that could be found in the literature (e.g. [10,19]) were used to develop a first version of the checklist with 22 items.

Before the study, it was performed a small pilot study with students from another course in order to evaluate the checklist and, possibly, to improve it. The most important issue pointed out by the students was some lack of clarity about what was intended with each question. To improve clarity it was added some guidance to each question of the checklist and its organization changes.

The 22 items are the following:

1. Are references to other requirements correct? Guidance: In case in a use case there is any reference to another use case this should be correct.
2. Has a priority been set for each requirement? Guidance: Whether the priority of each requirement has been defined and whether both the requirements table and the requirements descriptions are sorted by priority.
3. Is the definition of priority judicious? Guidance: If the priority is based on importance and difficulty.
4. Is the requirements detail adequate and consistent? Guidance: Whether the requirements are suitable to be used later in the next phase of design.
5. Specification is complete? Guidance: All known customer needs are included.
6. Are the anomalous (error) conditions that can be predicted documented? Guidance: These situations will be described in the alternative scenarios contained in the descriptions of the use cases.
7. Are there requirements that conflict with other requirements (how many)? Guidance: For example, contradictory requirements.
8. Is the language used in the requirements clear, concise and unambiguous? Guidance: Do you understand, refer only to what matters and do not raise doubts about what is intended?
9. Is the terminology consistent with that of the problem? Guidance: The terminology must be identical to that of the problem.
10. Do the requirements not present Portuguese errors ? Guidance: For example: spelling and grammatical errors.
11. Are the requirements within the scope of the project problem? Guidance: Requirements must not exceed the limits of the desired system.
12. Are the scenarios described correctly? Guidance: The order of the event stream is correct, and you can see who starts the events, what the inputs, outputs, and the processing are.
13. Are requirements testable? Guidance: If there are objective tests for each requirement.
14. Are there redundant requirements? Guidance: If a requirement is duplicated, or can be obtained from another part of the requirements.
15. Is each requirement uniquely identified? Guidance: The identification of each must be unique.
16. Is each requirement really a requirement? Guidance: A requirement should not concern design or implementation, i.e. other phases.
17. Is the documentation compliant? Guidance: If you agree with the standards of the documents defined for the project and the disciplines involved.
18. Do the requirements seem to be achievable in the time stipulated? Guidance: If the time set for the implementation of each requirement is realistic.
19. Conceptual diagram is complete? Guidance: For example, if there are actors that did not match classes and should, or if there are names in use cases that do not appear as classes and that should also.
20. Conceptual Diagram does not present unjustified classes? Guidance: You should not have classes outside the scope of the problem domain.

21. Are class relations correct? Guidance: To evaluate the type of association and its multiplicity.
22. Are the classes clear? Guidance: Is it understood the role of the class and its name (and attributes) is suggestive?

For each item in the checklist it was necessary to add 3 columns. One column to allow the participants to classify the SRS quality for each item. The classification scale defined was from 1 to 5. A second column for annotation of the number of defects (violations) for each item and a third to specify those defects were also added in order to gather more complete data. The instrument can be seen as a questionnaire.

3.5 Variables and Measures

The research questions involve two main concepts: quality of the reviews performed by the groups; and the quality of the groups. We start by analyzing the latter. Before the exercise each group received a mark for its SRS document (TM - teacher mark) given by the teachers. This is a numeric value between 0 and 20. The TM, besides being used to classify each SRS document, it is also a measure of the quality of each group (GQ - group quality).

During the exercise, each group should produce a grade for the SRS document they review (GG - group grade). For the exercise it was decided to not overload teachers with additional work, which implied that the validations made by the groups were not marked by the teachers. Without such information, there was no direct measure of the quality of each group review (RQ - review quality).

However, it is reasonable to expect that for a good review, the grade attributed by a group to a SRS would not be very distinct from the mark produced by the teachers to the same SRS. This can be seen as an indicator of the quality of the review. Assuming this, RQ was evaluated in the study through the proximity between the mark given by the teachers to the SRS (TM) and the grade attributed by the group to the same SRS (GG). More specifically, the proximity is computed as the absolute difference between the two measures.

3.6 Hypothesis

The original research questions were divided into statistical hypotheses as follows.

RQ1. To answer this question it was decided to see if there was a correlation between the grades attributed by the groups (GG) and the marks attributed by the teachers (TM) to the same SRS document.

$H1_0$ The grades attributed by the groups and the marks attributed by the teachers are independent. That is, TM and GG are mutually independent, $\rho = 0$.

$H1_a$ There is a correlation between the grades attributed by the groups are independent of the marks attributed by the teachers. That is, $\rho <> 0$.

RQ2. To answer this question it was determined to use the correlation between the mark received by the validation group for its SRS document (GQ) and the quality of the review ($RQ = abs(TM - GG)$).

- $H2_0$ The quality of the reviews are independent of the quality of the groups. That is, RQ and GQ are mutually independent, $\rho = 0$.
- $H2_a$ There is a correlation between the quality of the reviews and the quality of the groups. That is, $\rho <> 0$.

The group classifies the SRS document according the items of the checklist using a scale 1 to 5. To obtain a unique value for the GG it will be used the median of the classification given to each item. TM is a continuous variable in the range 1 to 20, so GG was first normalized to that range.

For both research questions, the main interest is to see if relationships are monotonic. To that, it was decided to use the Spearman coefficient, with an alpha of 5% for two-tails.

3.7 Process

The process for this study involving participants inspect SRS documents possess the following steps:

1. Selection of participants.
2. Explanation of the exercise to the teachers.
3. Explanation of the exercise to the team leaders.
4. Conduct the exercise in the class with teacher supervision.
5. Gathering and analysis of the results.

3.8 Preparation and Execution

Before the exercise, there was a meeting with all teachers about the exercise. Later, but before the exercise start, the team leaders met with the course responsible. At this meeting, the course responsible presented the exercise, its goal and the procedure to be followed during the exercise. The students became aware that they need to review a SRS from another group.

In order to guarantee the anonymity of the authors, the validation groups must be not aware of the identification of the SRS authors, their colleagues. To achieve this, previously to the study, each group was told to submit two versions of their reports: one report with identification and another without groups's identification. To each of latter documents, it was given a random number, so they could be identified later. All SRS documents were verified about their anonymity. After that they were uploaded to a server from where they could be read but not downloaded.

At the beginning of the exercise the teachers explained briefly to the students what should be done by the teams. After that teachers supplied the teams with the questionnaire (checklist) and the URL of the SRS document the team should review. The questionnaire was supplied in paper and no tool was used to help in the validation task. Because it was not possible to download the document, then each group read it online.

Each group inspected the respective SRS document and classified it according the checklist points giving a classification between 1 and 5 to each point. The team leader was responsible for filling the checklist with the group classifications and return it to the teacher. Group classifications to each checklist item where given by consensus. After the exercise the teacher delivered the filled checklists to the course responsible.

The exercise had an indicative time for completion of one hour but it was not rigid. Soon it was verified that one hour was not enough and some groups extend it to nearly two hours, but the time required by each group was not registered by the teachers.

As a quality measure, teachers were instructed to classify the group commitment to the exercise. After the exercise, several teachers showed many doubts about their classification.

3.9 Results

Forty five (groups) filled the questionnaire. Two questionnaires were removed because the respective team leaders did miss the kick-off meeting. The result was forty three (43) answers to be analysed. Also the third and forth columns of the checklist could provide relevant information, some were incomplete and many present several discrepancies in the way they were filled. Although more correlations and information would be interesting to obtain using those columns, they have to be ignored.

For each group it was determined the median and mean of their answers to each item of the questionnaire. The median and mean of those medians and means is showed in Table 1. All computations were done using the R package. For both statistical tests was used the *cor.test* function.

Table 1. Overall median and mean.

Groups	Median	Mean
43	4	3.7482

Regarding **RQ1**, Spearman's rank correlation was $\rho = 0.3844098$ and $p\text{-}value = 0.01092$. So, since $p\text{-}value$ is less than the defined significance of 0.05, the null hypothesis for $RQ1$ was rejected.

About **RQ2**, to obtain the value for the review quality (RQ), firstly was computed the mean of the classifications (1–5) that each group attributed to each questionnaire item ($GGmean$). The computation of the mean was done despite the problem with scales because this is an exploratory study. Next, the value was normalized to the range 0–20, $GGmean20$. Finally, it was computed the absolute value of the difference between TM and $GGmean20$.

Spearman's rank correlation was $\rho = -0.04078396$ and $p\text{-}value = 0.7951$. According the obtained $p\text{-}value$ for a the significance level of 5%, the null hypothesis for $RQ2$ could not be rejected.

3.10 Analysis and Discussion of Results

Regarding the first research question, the observed correlation between the teachers' marks and the groups' marks is statistically significant, but not high. However the criteria used by the groups and the teachers were not the same. The teachers didn't follow the students' checklist, but all follow the same set of criteria. So, the question was to know if the correlation would be higher if criteria were the same. If higher, it would give more confidence on peer reviews and on their use for peer assessment.

The result of the second research question was negative, that is, the correlation between the quality of the groups and their reviews as was measured was not statistically significant. Possibly, if instead it was used the mean of the marks of the students during the program and not simply of their SRS, the result would be different. Furthermore, the quality of the reviews was measured as the distance between the groups' grades and the teachers marks. This is not a direct measure and could have had some effect on the results.

As seen, the checklist covers quality criteria for traditional requirements specifications. It is relevant to know its adequacy to agile development. A recent study [2] about agile requirements specifications investigated 16 papers and found 28 different quality criteria used on those papers. The authors grouped the criteria, if cited by 3 or more papers, in three global criteria: completeness, uniformity and, consistency and correctness. Uniformity is about the adherence to some specific agile format for the specification. Completeness, should be interpreted in an agile context, that is, as just-enough and just-in-time requirements, where requirements are incomplete. The three global criteria are more detailed in [20]. Looking to our checklist, can be seen that in general it covers those quality criteria, except for uniformity, which is specific to agile. This means that the checklist can be easily adapted to agile development with smaller sets of requirements.

Anyway, it is necessary a more in-depth analysis to align the checklist with the quality criteria used in agile projects. Currently, and contrary to the time of the study, the project is supported by a project tracking tool that allows the course responsible to keep track of the work performed by the groups.

For the study it was used a large set of requirements. The difficulty to validate a large set of requirements would be higher than for a small set, but, at the same time, it is expected to better prepare the students to validate requirements specifications of any size and, in this way, using any development methodology. As stated, most of the requirements have been elicited at an early stage of the project, but new requirements have been added at each new iteration. However, the study refers only to specification documents submitted with the initial requirements. Anyway, this means that the methodology used can be considered hybrid.

3.11 Threats to Validity

The validity of measures used can not be regarded as guaranteed, in particular, the review quality (*RQ*) and group quality (*GQ*) measures. It is not obvious the possible negative impact on validity of *RQ* because it was observed some correlation between groups' grades and teachers' marks, the two measures *RQ* depends. The *GQ* is equal to the teachers' marks to the same SRS the group reviewed. The use of columns 3 and 4 of the checklist would help to improve the validity of the results, but, as explained, their use was not possible.

The criteria used in the teachers' assessment of the SRS documents were distinct from the criteria used by the groups. Teachers' assessments also incorporate pedagogical dimensions (as the report presentation, or the formalism of the UML diagrams) and did not use the checklist. There is some percentage of common criteria but we were not able to achieve a value for that. We speculate that the different criteria could have been a main bias factor for the unobservable correlation between *RQ* and *GQ*. So, in future research we intend to assure common criteria. The large number of teachers involved could also affect the correlations between their marks and the students grades.

Another threat has to do with the use of the supplied SRS template because not all groups respected completely this template, which should have affected the SRS document presentation and the corresponding reading by the validation groups. Also, it should had some impact on the examination by teachers. Because each group possess their own view of the problem, it is possible some bias when evaluating others SRS.

The time indicated initially for the exercise was not enough for all groups. Although this could apparently had some impact on the quality of the answers, it was overcome by the fact that teachers gave additional time to the groups.

We are aware that students commitment to the exercise could have an important impact on the results. However, we have no reliable measure about students commitment to the exercise and, consequently, the possible impact on the performance of the several groups. Possibly the teachers that supervise the exercise could also have an effect on the groups' performance but all validations were done in identical circumstances, with the presence of a teacher.

In [21] were found some differences between students (freshman and graduates) due to their experience. As stated in Sect. 3.3 a small number of students possess industrial experience in software development. However, we did not identify such situations.

Also, the grouping and ordering of the questions in the checklist may not be the most correct. Possibly, it should start from the more general and substantive to the detailed an less substantive questions. For example, questions like the 5th and the 11th could be placed in the first positions. This needs to be further investigated.

4 Conclusions

This paper presented a study of an exercise performed during regular classes where groups of students reviewed requirements specifications developed by their peers. In the study it was observed a correlation with statistical significance between the grades attributed by students and the marks given by the teachers to the same requirements specifications. That correlation was not high. So, peer reviews can be considered reliable to some extent. This means that teachers can use those reviews as a form of peer assessment to aid them in marking students work, but carefully. This was the main contribution of this paper.

At the same time, the study found no correlation between the quality of students' groups and the quality of the requirements reviews they do. However, indirect measures to classify the quality of the groups and of their reviews were used. Researchers that intend to develop similar studies should pay attention to guarantee that criteria used by teachers and students to evaluate their requirements specifications, or other relevant artefacts, are as identical as possible.

The software development model used on the exercise can be classified as hybrid, that is, using weekly iterations, but with a strong emphasis on requirements engineering up front. With such approach, students are faced with a large amount of requirements to validate. This should have increased the difficulty of the exercise, turning it more challenging to the students, which was considered positive for training purposes. The current increase in the use of hybrid models by the industry, as stated before, makes this kind of studies more relevant.

The effectiveness of reviews in software development is well documented [22] and the use of checklists in inspections is considered as having a positive effect in requirements engineering in agile projects (e.g. [23]). As shown, the checklist used by the groups requires some changes to be suitable for use in pure agile projects, including the use of tools, as done in modern code reviews. But using tools should not avoid having the knowledge underlying performing validations. From the limitations presented, this research should continue by improving the checklist, the measures and, if possible, its integration with tools to be in complete line with a modern requirements review perspective.

Acknowledgment. The authors would like to thank their colleagues and students.

References

1. Ambreen, T., Ikram, N., Usman, M., Niazi, M.: Empirical research in requirements engineering: trends and opportunities. Requirements Eng. **23**(1), 63–95 (2016). https://doi.org/10.1007/s00766-016-0258-2
2. Heck, P., Zaidman, A.: A systematic literature review on quality criteria for agile requirements specifications. Software Qual. J. **26**(1), 127–160 (2016). https://doi.org/10.1007/s11219-016-9336-4

3. Boehm, B.W.: Software Engineering Economics, 1st edn. Prentice Hall PTR, Upper Saddle River (1981)
4. De Lucia, A., Qusef, A.: Requirements engineering in agile software development. J. Emerg. Technol. Web Intell. **2**(3), 212–220 (2010)
5. Kuhrmann, M., et al.: Hybrid software development approaches in practice: a European perspective. IEEE Softw. **36**(4), 20–31 (2019). https://doi.org/10.1109/MS.2018.110161245
6. He, L., Carver, J.C., Vaughn, R.: Using inspections to teach requirements validation. CrossTalk: J. Defense Softw. Eng. **21**(1) (2008)
7. Gigante, G., Gargiulo, F., Ficco, M.: A semantic driven approach for requirements verification. In: Camacho, D., Braubach, L., Venticinque, S., Badica, C. (eds.) Intelligent Distributed Computing VIII. SCI, vol. 570, pp. 427–436. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-10422-5_44
8. Loftus, C., Thomas, L., Zander, C.: Can graduating students design: revisited. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, SIGCSE 2011, pp. 105–110. ACM, New York (2011). https://doi.org/10.1145/1953163.1953199
9. Topping, K.: Peer assessment between students in colleges and universities. Rev. Educ. Res. **68**(3), 249–276 (1998). https://doi.org/10.3102/00346543068003249
10. Anda, B., Sjøberg, D.I.: Towards an inspection technique for use case models. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, pp. 127–134. ACM (2002)
11. Lahtinen, J.: Application of the perspective-based reading technique in the nuclear I&C context: CORSICA work report 2011. VTT Technical Research Centre of Finland, No. 9. VTT Technology (2012). http://www.vtt.fi/inf/pdf/technology/2012/T9.pdf
12. Basili, V.R., et al.: The empirical investigation of perspective-based reading. Empirical Softw. Eng. **1**(2), 133–164 (1996). https://doi.org/10.1007/BF00368702
13. Atoum, I.: A scalable operational framework for requirements validation using semantic and functional models. In: Proceedings of the 2nd International Conference on Software Engineering and Information Management, ICSIM 2019, pp. 1–6. ACM, NY (2019). https://doi.org/10.1145/3305160.3305166
14. Nazir, S., Fatima, N., Chuprat, S.: Modern code review benefits-primary findings of a systematic literature review. In: Proceedings of the 3rd International Conference on Software Engineering and Information Management, ICSIM 2020, pp. 210–215. ACM, NY (2020). https://doi.org/10.1145/3378936.3378954
15. Runeson, P., Host, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering: Guidelines and Examples. Wiley, Hoboken (2012)
16. Laitenberger, O., Atkinson, C., Schlich, M., Emam, K.E.: An experimental comparison of reading techniques for defect detection in UML design documents. J. Syst. Softw. **53**(2), 183–204 (2000). https://doi.org/10.1016/S0164-1212(00)00052-2
17. Rowlands, K.D.: Check it out! Using checklists to support student learning. Engl. J. **96**(6), 61–66 (2007)
18. IEEE-SA: IEEE Std 830-1998 - Recommended Practice for Software Requirements Specifications, IEEE-SA (1998). https://doi.org/10.1109/IEEESTD.1998.88286
19. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley Publishing, Hoboken (2009)
20. Heck, P., Zaidman, A.: Quality criteria for just-in-time requirements: just enough, just-in-time?. In: 2015 IEEE Workshop on Just-In-Time Requirements Engineering (JITRE), pp. 1–4 (2015). https://doi.org/10.1109/JITRE.2015.7330170

21. Runeson, P.: Experiences from teaching PSP for freshmen. In: Proceedings of the 14th Conference on Software Engineering Education and Training, CSEET 2001, pp. 98–107. IEEE Computer Society, Washington, DC (2001)
22. Jones, C.: A Guide to Selecting Software Measures and Metrics. CRC Press, Boca Raton (2017)
23. Kalinowski, M., et al.: Supporting defect causal analysis in practice with cross-company data on causes of requirements engineering problems. In: Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track, pp. 223–232. IEEE Press (2017)