IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 3, 2021, accepted December 2, 2021, date of publication December 13, 2021,
date of current version December 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135245

Evaluation of Critical Thinking in Online Software Engineering Teaching: A Systematic Mapping Study

OSCAR ANCÁN BASTÍAS[1], JAIME DÍAZ[1], AND CRISTIAN OLIVARES RODRÍGUEZ[2]
[1]Depto cs de la Computación e Informática, Universidad de la Frontera, Temuco 4811230, Chile
[2]Instituto de Informática, Universidad Austral de Chile, Valdivia 5110566, Chile

Corresponding author: Jaime Díaz (jaimeignacio.diaz@ufrontera.cl)

This work was supported in part by the Certificación de Educador Internacional de Ingeniería (CEII) Programme, Instituto InnovaHiEd, accredited by the International Society for Engineering Pedagogy (IGIP); in part by the Universidad de La Frontera under Project DID20-0012 and Project DI21-0016; in part by Project Innoving 2030 of Universidad Australia; and in part by the Centro de Estudios en Ingeniería de Software (CEIS).

ABSTRACT Critical thinking consists in analysing and evaluating the coherence of reasoning. This ability is crucial when we talk about software quality (SQ). SQ is closely related with the engineer's ability to judge and discriminate between solutions correctly, so students are required to analyse, evaluate and draw conclusions. Critical thinking, therefore, becomes a crucial part of the training of software engineers. The problem arises from the diversity of proposals and the lack of rigour in existing experiences, making it difficult to find specific recommendations, especially in online contexts. This article reports a systematic mapping study (SMS), the purpose of which was to detect, organise and characterise specific dimensions in online teaching-learning of critical thinking for software engineering. Based on the results of the SMS, we propose a preliminary framework for the evaluation of critical thinking in the training of software engineers in a context of online higher education. It is expected that this proposal will serve as a basis for instructors of the discipline when evaluating critical thinking in a context of online teaching.

INDEX TERMS Critical thinking, online education, software engineering education.

I. INTRODUCTION

The ubiquity of information has resulted in an unprecedented dependence on technology in our society [1]. As a consequence, the demand of the IT industry for engineers capable of creating high quality software has increased [2], [3]. Software quality is closely related with the engineer's ability to judge and discriminate between solutions correctly, as well as adapting to the continuous changes in tools and techniques. This implies maintaining a high level of critical thinking in every decision during software production [4].

Critical thinking is considered an essential skill in the training of future engineers and generally in the 21st century [5]. However, studies investigating critical thinking in students on courses related with software production are limited, even though it is a competence required in the discipline [6], [7]. Likewise, the instruments for evaluating critical thinking with the greatest cover are generic and standardised [8], and do not consider the disciplinary context or the mode of study. The scarce evidence available is insufficient to establish a consensus on how to evaluate the components of critical thinking in students on courses related with software production, leading to contradictory results [9]–[12].

In this scenario, the need arises to carry out a review of the current state of the evaluation of critical thinking in software engineering training imparted online, to give teachers and students a guide as to which are the most important elements to consider and what gaps in knowledge need to be addressed.

This article presents a systematic mapping study of the literature (SMS), carried out by applying the systematic mapping protocol proposed by Petersen [13]. Reviewing and analysing hundreds of research articles enabled us to establish the main topics currently being addressed in the discipline, and to identify missing areas and recent trends.

The principal contribution of this article, therefore, is to determine the central themes that need to be considered for the application of any strategy for teaching critical thinking in software engineering, in an online environment.

The article is organised as follows: Section II describes the main characteristics of the study; Section III presents the method used for systematic mapping; Section IV describes

The associate editor coordinating the review of this manuscript and approving it for publication was Mark Lee.

the main results; Section V offers answers to the Research Questions; Section VI, based on the results obtained, proposes a preliminary framework for evaluating critical thinking in the context of the teaching of software engineering online; Section VII summarises the biases of the work; and finally Section VIII presents the conclusions of the study.

II. BACKGROUND

A. CRITICAL THINKING

Many of the current definitions of critical thinking in the context of education are based on Bloom's taxonomy, in which the three highest levels of the hierarchy generally represent critical thinking: analyse, evaluate and create [14]. This has led to a variety of definitions with no apparent consensus between investigators, resulting in ambiguities when teaching and when carrying out practical assessments [15]. Nevertheless, there is an agreement that the skills of critical thinking consist principally in analysing arguments, drawing inferences, judging or evaluating, and taking decisions, with the object of guiding problem-solving [16].

Furthermore, critical thinking can be approached as a way of being – thought; or of acting – willingness [17]. Willingness to employ critical thinking is the intrinsic motivation of the subjects to value, consciously and actively, the arguments presented [18]. The most important aspects of willingness are related with mental openness, curiosity, the desire to be informed and respect for external or different points of view [16].

Previous or specific knowledge of a domain play a central role, both in the skills of critical thinking and in the willingness to employ it, as they allow us to establish the topics to which critical thinking is to be applied [19].

Together, the skill of thinking critically, the willingness to do so, and previous knowledge about a topic, represent the central structure for establishing a teaching-learning strategy for critical thinking.

The commonest strategies in use today are fusion and immersion. In [20] the authors propose the fusion approach, which implies in-depth instruction in the material, plus separate instruction in critical thinking [21], as well as including everyday problems with the idea of teaching the students how to transfer the skills of critical thinking to specific contexts [22]. The immersion approach assumes that the students acquire critical thinking skills as a consequence of learning the course contents, and not as independent parts of the course [23].

B. EXPERIENCES OF TEACHING CRITICAL THINKING ONLINE

The literature reports various teaching proposals that show a positive impact on critical thinking, mainly following the immersion approach: by problem-based learning [24], technological learning [25] or flipping part of the contents [26]. However, in a completely online environment, but with a face-to-face culture in both students and teachers, this becomes a serious challenge [27].

Recent experiences indicate that it is possible to address the development of critical thinking in an online format with relative success using project-based teaching [28] and inquiry-based learning [29] to develop the willingness to employ critical thinking [30].

A critical thinking teaching model based on web simulations has also been proposed; it received positive evaluation when trialled in students on the Network and Communications course. This model includes a simulation process which starts with visual inspections, combined with online discussion to encourage socio-constructivist learning. It has been shown to have a significant impact on the acquisition of critical thinking in system design, assessed by pre and post-testing [31]. A different model of teaching design for active learning in online contexts has been proposed, which encourages critical thinking in problem-solving. In this model the teacher guides the reflective process of analysis, and designs learning activities which contribute to active, student-focused learning, establishing situations of critical analysis [32].

In [33], the authors report that Japanese and German students collaborated in a training programme in software engineering through learning based on collaborative projects online. Incorporation of an intercultural dimension produced a positive result in the framework of this teaching strategy, although it is more demanding for both students and teachers. Collaborative work proved to be a positive strategy for the development of critical thinking. A similar initiative in a computer sciences course applied a model of encouraging collaboration and the evaluation of online teamwork, based on evidence of constructive feedback between peers. This model showed that peer comments in the analysis of complex problems contribute to teamwork and improve the critical skills of the participants. Thus encouraging discussion emerges as a strategy for reflection on both individual and team performance [34].

C. EVALUATION OF CRITICAL THINKING IN THE ONLINE CONTEXT

Evaluating critical thinking skills and/or the willingness to employ critical thinking presents great challenges to both teachers and students. In this context, problems arise related with the reliability and validity of the instruments available, stressing that traditional evaluation formats are not appropriate for measuring critical thinking skills [16].

Since the aspects included are drawn mainly from the three levels of Bloom's taxonomy, creativity is an inherent element of these three levels. Furthermore, evaluation of this aspect introduces subjectivity and error, as Silva points out [21]. There are other approaches that suggest the use of open question problems, rather than multiple choice [35]. By the same token, the problems set should have more than one possible solution, allowing students to argue more than one point of view [36].

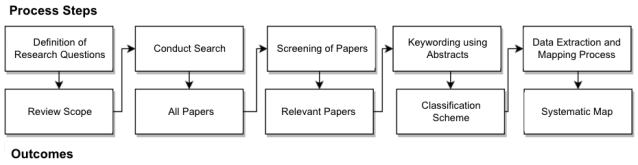

FIGURE 1. Stages of the systematic mapping process.

Standardised instruments have been used to enable participants to assess their own critical thinking; one of these is CAT (Critical Thinking Assessment Test) [37], proposed jointly with project-based teaching to integrate critical thinking by solving complex problems [9]. This experimental study, with a control group, showed evidence of an improvement in critical thinking skills among students using an active, project-based teaching strategy. Nevertheless a comment was made with respect to the low motivation of the students who were asked to answer the same test twice, which may have affected the results [8].

In other models for teaching software engineering, personalised instruments have been established for evaluating critical thinking. Integration of a model to encourage critical reading, SQ4R [38], has been used as an analysis strategy for software engineering. Students perceive this strategy as a useful tool for improving comprehension of course contents and for the effective development of software engineering projects [6]. Another work proposes standardised evaluations combined with observation and qualitative feedback from the participants, identifying opportunities for improving critical analysis [8]; then there are studies which include only personalised instruments for evaluating this skill, based on authentic learning [39], thought-based learning [40] and the generation of evidence of learning by means of formal marking systems or observation parameters [41].

Consequently, critical thinking is seen to be vitally important for software engineering students in higher education, since various techniques and solutions must be considered at every stage of the development process. Nevertheless, so far as we know there has been no evidence of proposals for how to link and evaluate explicitly critical thinking skills with the competences required for developing high quality software, in virtual contexts and in the new scenarios of collaborative learning.

III. METHOD

The systematic mapping technique proposed by Petersen [13] offers a way of verifying, analysing and categorising results related with a specific topic or area of interest, thus allowing the scope of the investigation to be determined, and the knowledge obtained to be classified.

Performing a systematic mapping study (SMS) involves following the stages described in Fig. 1 sequentially. Thus as the various stages of the process are completed, concrete results are obtained which form the direct input of the following stage, in order to achieve systematic mapping as the final result.

TABLE 1. Research questions.

ID	Research questions	
	Item	Object
RQ1	What research topics characterise the evaluation of critical thinking in software engineering teaching?	Determine the topics related with the evaluation of critical thinking in software engineering teaching.
RQ2	What topics related with the evaluation of critical thinking have been published in the field of software engineering teaching?	Identify the topics related with the evaluation of critical thinking have been published in the field of software engineering teaching
RQ3	What is the predominant method used in software engineering teaching to evaluate critical thinking?	Identify the predominant method used in software engineering teaching to evaluate critical thinking

TABLE 2. Categories and key words.

Category	Key word	Variants
Evaluation Framework	Assessing framework	Assessing guideline; Assessing approach; Assessing method; Assessing framework; Assessing strategies; Assessing model; Assessment
Critical Thinking	Critical Thinking	Critical Thinking; critical thinking skills
Software production	software development	software developing; software development; software build; software production; software engineering
Online Mode	Online Higher Education	Online learning; Online Courses; Remote Training; Online Training; Remote Education; Remote Teaching

The activities that make up the systematic mapping process are described in the following sections.

A. DEFINITION OF THE RESEARCH QUESTIONS

The first stage is the presentation of the Research Questions (RQ) which provide the methodological basis of the rest of the process. Table 1 details the three RQ of this investigation.

B. SEARCH EXECUTION

The first step towards obtaining results was to define key words for the study in English, with possible variants; these were obtained from the RQ (see Table 2):

The final search chain used was as follows:

((“assessing” OR “assessment”) AND (“guideline” OR “approach” OR “method” OR “framework” OR “strategies” OR “model”)) AND (“critical Thinking” OR “critical thinking skills”) AND (“software developing” OR “software development” OR “software build” OR “software production” OR “software engineering”) AND (“online learning” OR “online courses” OR “remote training” OR “online training” OR “remote education” OR “remote teaching”)

In selecting data sources we included our own search engines, with complete access from the site where the study was carried out. The sources in which the search was carried

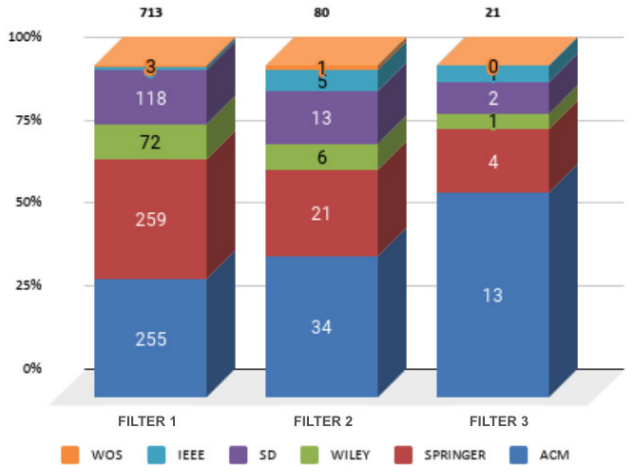

FIGURE 2. Study sources by filter.

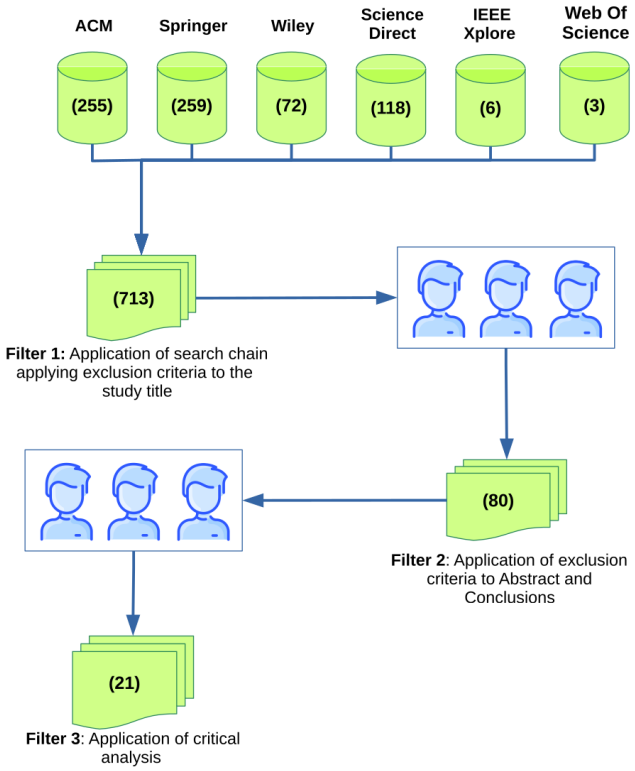

FIGURE 3. Article filtering stages.

out were ACM, Springer, ScienceDirect, IEEE Xplore, Wiley and WOS. The search was performed in January 2021. Fig. 2 presents a summary of the results.

C. ARTICLE SELECTION

The articles were selected by applying inclusion/exclusion criteria to the results delivered by each search engine (see Fig. 3).

- Inclusion: Proposals, strategies, tools, experiments, methodologies, artefacts and/or models that support the evaluation of critical thinking in software engineering teaching in an online context.
- Exclusion: Studies not related with software engineering; programming or technology applied in a traditional environment. Literature reviews.
- Data range: 2011 - 2021.

Research works were selected first by the inclusion criterion through analysis of the title, abstract and key words; this produced a large number of works that make a significant contribution to the study area. At the same time the date range filter was applied and duplicates were eliminated.

In the following step, the exclusion criterion was applied to the abstract of the article, eliminating short articles (less than 3 pages), theses, technical reports and tutorials. Only articles focusing on the evaluation of critical thinking and software development teaching were retained. Finally, a full-text analysis was carried out of the candidate documents, selected individually by secret vote of all the authors. The concordance between the researchers was validated using Fleiss' Kappa Index, as proposed by Gwet [42], which gave a reliability of 87.1%. The final product was 21 relevant articles for the analysis.

D. REPLICABILITY

We provide the complete data set[1] used to perform analyses, in order to replicate and validate our study.

IV. RESULTS

The first group of articles selected were literature reviews; four articles were identified that addressed similar topics to that of the present proposal. Although none of them addressed exactly the topic of critical thinking in software engineering teaching in online contexts, they do represent an interesting basis for the study:

Chanin *et al.* [43] identify important contributions of education in engineering in a context of ''software start-up''. They describe a variety of best practices and methodologies particularly suitable for businesses. In this work we can recognise the emergence of skills linked to critical thinking.

Veras *et al.* [44] found that activities in online classes follow three main strategies: (1) project-based learning (38.3%.); (2) problem-based learning and self-teaching (50.0%.); and (3) team learning (7.7%). The studies reviewed report challenges such as teachers with work overload and time limitations, as well as difficulty in maintaining student motivation in hyper-mediatised environments. This context has a negative impact on the willingness to think critically.

Garousi *et al.* [45] report a meta-analysis aimed at consolidating the alignment of education in software engineering with the needs of the industry. Their results lead to the following conclusions: (1) software requirements, design and testing are the most important skills; and (2) the biggest knowledge gaps are found in model-management and the processes of software engineering, design (and technological architecture) and testing.

[1] https://doi.org/10.5281/zenodo.5773112

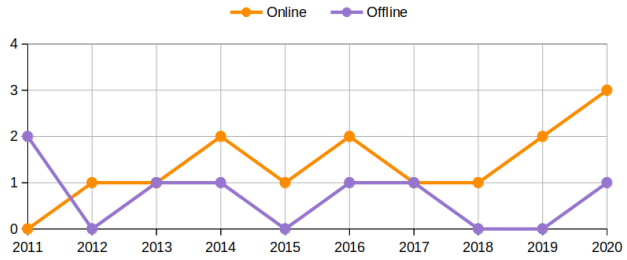

FIGURE 4. Evolution of publications per year.

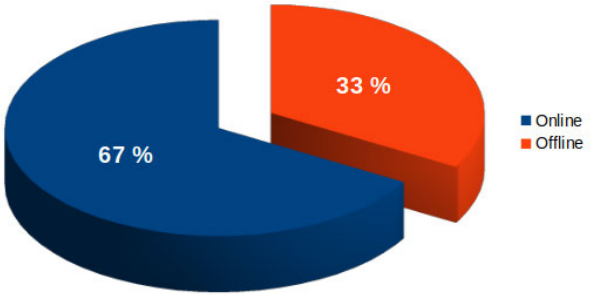

FIGURE 5. Percentage of Online/Offline modes.

Finally, Anthonysamy *et al.* [46] compiled information on metacognitive knowledge, resource management and motivational belief strategies. The authors conclude that Self-regulated Learning Strategies (SRLS) have a positive correlation with non-academic results. These strategies are closely linked with a willingness in students to employ critical thinking.

Positive evolution and growing interest can be seen over the ten years included in this study. One particular aspect is the considerable increase in the number of studies related with virtual education in software engineering. This may have been promoted by the context of the COVID-19 pandemic and the quarantine processes applied in educational institutions as a result (see Fig. 4).

Fig. 2 reports the sources of the data. The largest numbers of articles came from Springer and ACM. As the filters were applied successively, all the sources were reduced by a similar percentage, and most of those selected (81%) were obtained from these two sources.

Finally, 38% of the results were drawn from journals, the remainder were articles from scientific conferences.

Fig. 6 shows the classification of the different categories emerging from the analysis performed in this study. Eight categories were identified:

- **Software engineering (SE) teaching strategies:** The various types of initiatives applied in the teaching-learning processes are described, such as: Project-based learning, Problem-based learning or Flipped classroom.
- **Mode**: Defines the format of the intervention or proposal, divided into ''offline'' (in person) and ''online'' (remote).
- **Type of study**: Defines the methodology applied to the evaluation of the results.
- **Context**: Determines the scope declared by the authors. This ranges from global applications, through technology companies down to applications in local software production companies.
- **Evaluation of critical thinking**: Reports various initiatives to evaluate critical thinking, highlighting those constructed especially for experiments (ad hoc) and standardised solutions.
- **Type of contribution**: This analyses the results, reporting the types of conclusions reached by the authors, and how they are offered to the community. For example: recommendations, models, strategies, or relative findings (from experiments).
- **Stages of software development**: Describe the stages of the traditional software development cycle in which the topic of critical thinking is addressed.
- **Evaluation strategy**: Indicates which instruments were used to measure the evaluation of critical thinking. For example: marking systems, peer evaluation, self-evaluation, etc.

Finally:

- **Type of study**: 23.81% are ''Reports of experiences'', i.e. they present an approach without necessarily the rigour of a case report. 38.10% are ''Proposals'' of the authors themselves.
- **Teaching strategies**: 52.38% use (commercial) development exercises in local software production companies. In other words, although the experiments are performed in an online context, most of them are to solve local problems and respond to internal needs.
- **Evaluation processes in software engineering**: 42.86% do not define a direct strategy for measuring performance. This information is often omitted, and because it was outside the scope of the initiatives it was probably not considered important by the authors. Nevertheless, it may be noted that 71.43% of the studies do not declare any kind of evaluation of critical thinking in their experiments.

In the analysis by category, we note that 57.14% offer only relative findings, but have no clear evaluation mechanisms either for topics directly related with engineering or for teaching-learning processes.

The articles selected are listed in Appendix A. No single author or research group predominates. Figure 4 shows that the oldest articles date from 2011, while the most recent and most numerous are from 2020.

If we analyse the key words declared by the authors of the 21 articles selected, important findings may be noted (see Fig. 7). Leaving aside the obvious results ''software engineering education'' and ''online platforms'' (28.4%), two frequent key words are ''Problem-based learning'' (10.4%) and ''Project-based learning'' (9.4%). Indeed, nearly 60% of the results mention a collaborative approach based on projects or joint problem solving. It is important to note, however, that the majority of the articles describe practical experiences (with low methodological rigour), and that only

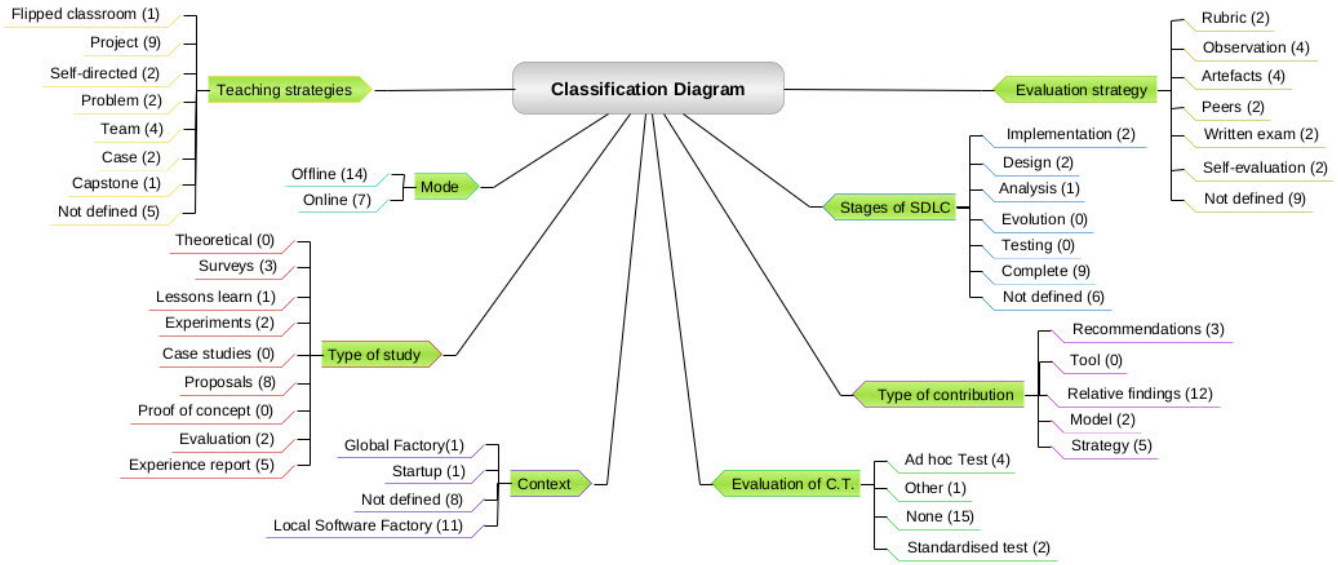

FIGURE 6. Article Classification Diagram.

FIGURE 7. Keywords from selected articles.

4.5% of the total propose meta-models for the teaching of critical thinking.

V. DISCUSSION

This section seeks to give answers to the Research Questions formulated in Section III, based on the results obtained and the systematic maps.

A systematic map is drawn based on simultaneous counting and classification of the dimensions of interest; the diameter of the bubble is proportional to the number of investigations linked to the study dimensions (Fig. 8, 9 and 10). This allows a general view of the field of study.

A. RQ1: WHAT RESEARCH TOPICS CHARACTERISE THE EVALUATION OF CRITICAL THINKING IN SOFTWARE ENGINEERING TEACHING?

Analysis of the articles revealed three central research topics: (i) Teaching strategies, (ii) Evaluation strategies, and (iii) Stages in the software development life cycle (see Fig. 8).

- **Teaching strategies**: approximately 42% of the articles addressed the investigation from the perspective of project-based learning, which was one of the most frequent strategies used to evaluate the development of critical thinking (see Fig. 11). Nevertheless, the great majority of the initiatives lacked any formal support for evaluating critical thinking. On the contrary, the "flipped classroom" strategy is little used, around 4%, and as in the rest of the strategies there was no mention of the use of instruments to evaluate thinking, nor of Use Scenarios for the software life cycle. The above may be attributed to investigations indicating that this type of teaching strategy presents a low level of compliance with pre-class activities [47], and a marked resistance if it is the first time that the course has been taught [48], which would have a direct impact on the arguments proposed.
- **Software engineering evaluation strategies**: A considerable number of articles (9), give no details of how learning was evaluated (see Figure 12). Approximately 40% of the articles are split between "Observation" and the development of "Artifacts" that are evaluated. This agrees with the difficulties of setting up evaluations in the context of software engineering teaching proposed in [49].
- **Stages in the software development life cycle (SDLC)**: There is a striking absence of studies which address stages like testing and/or software evolution exclusively, or the maintenance processes associated with this stage (see Figure 13). This offers a niche for study in the widely researched connection between tests and software quality, and how these elements are linked with critical thinking skills. We note that nine articles report the use of a complete life cycle, but without giving details of the stages, which makes it impossible to investigate exactly which tasks might have an impact on the student's critical thinking.

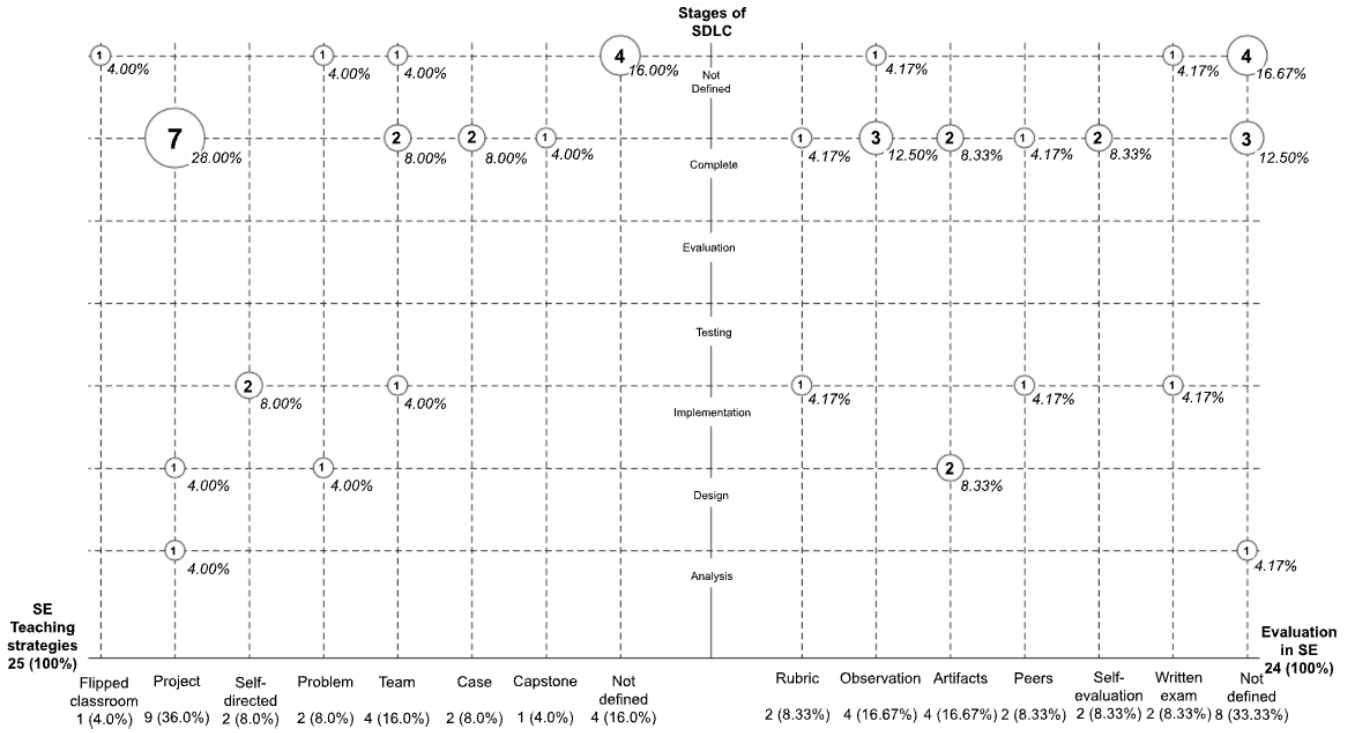

FIGURE 8. Distribution of studies on critical thinking over SDLC, evaluation strategies, and teaching strategies.

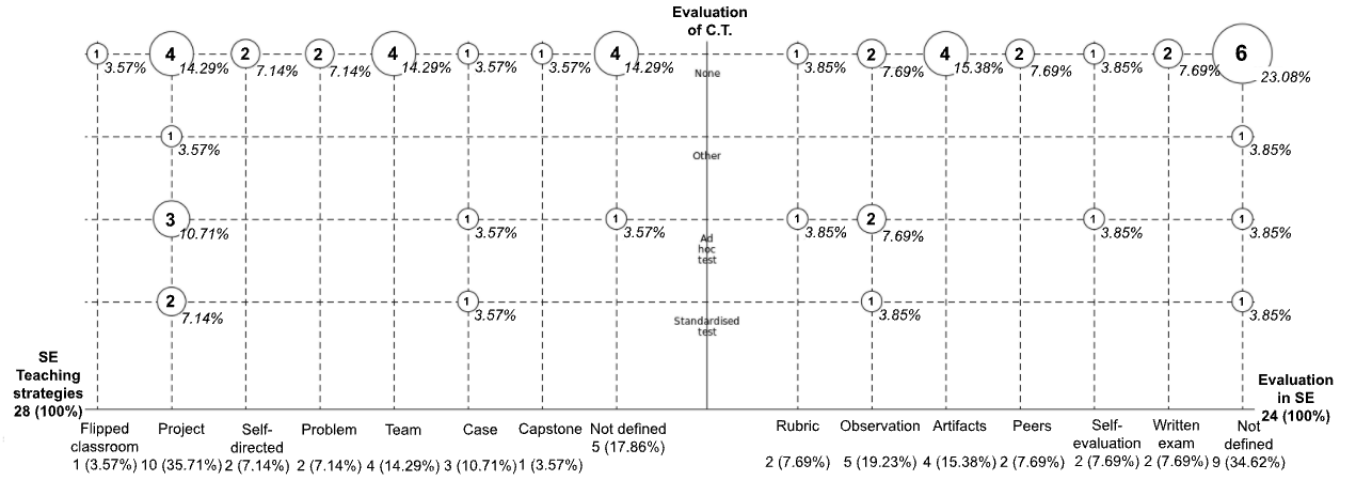

FIGURE 9. Distribution of studies on critical thinking over measurement artifact, evaluation strategies, and teaching strategies.

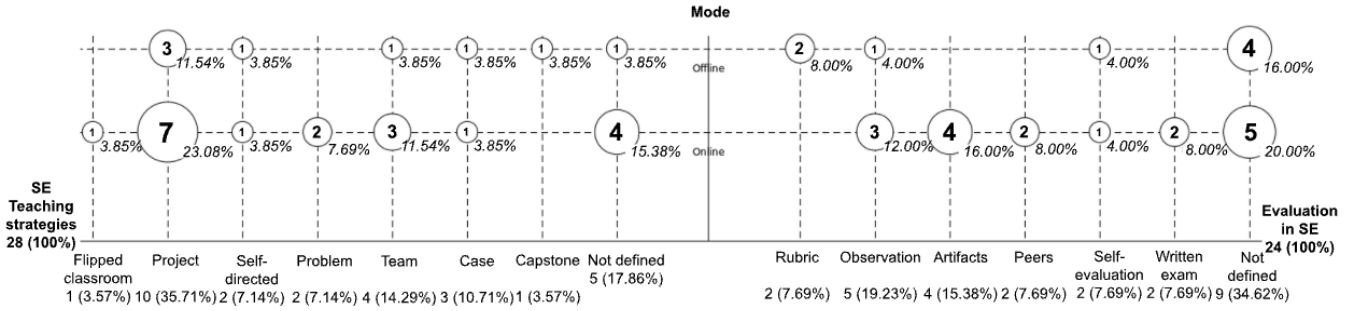

FIGURE 10. Distribution of studies on critical thinking over mode, evaluation strategies, and teaching strategies.

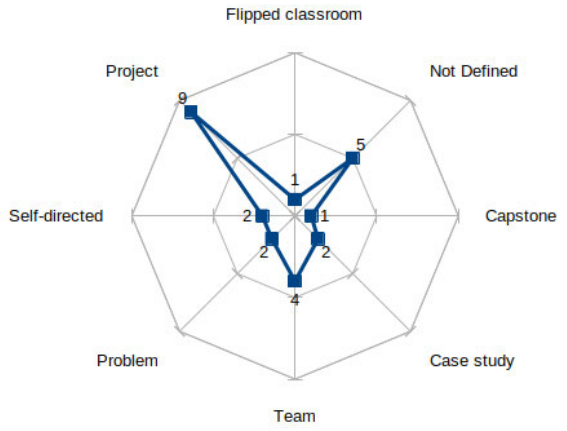

FIGURE 11. Software engineering teaching strategies.

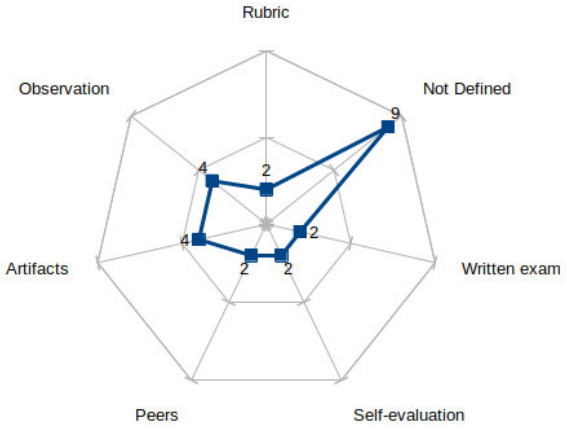

FIGURE 12. Evaluation strategies in the context of software engineering teaching.

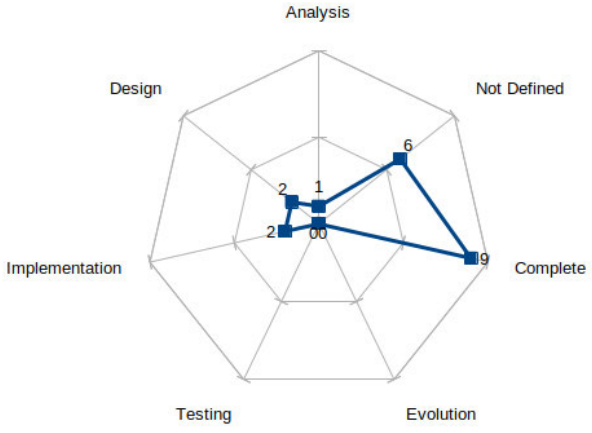

FIGURE 13. Stages in the software development life cycle.

B. RQ2: WHAT TOPICS RELATED WITH THE EVALUATION OF CRITICAL THINKING HAVE BEEN PUBLISHED IN THE FIELD OF SOFTWARE ENGINEERING TEACHING?

The articles reviewed report three main categories of instruments for measuring critical thinking:

- **Universal or generic evaluation instruments**, applied without specifying a discipline.
- **Ad hoc evaluation instruments**, i.e. adaptations or adjustments of existing instruments for measuring critical thinking.
- **Others**, i.e. instruments or artifacts developed specifically for the situation to be measured.

In this classification, 67.86% of the articles (see Fig. 9) do not report the use of instruments for evaluating critical thinking, despite mentioning it as a potential finding or element to be developed. This is unexpected, and reveals fundamental problems with the definition of the study, as mentioned in [16] and [50]; and also the lack of investigation into the evaluation of critical thinking in software engineering teaching environments [6].

Most of the other 32.14% fall into the test categories Ad hoc, Standardised and Other. These results may be related with the secondary nature attributed historically to critical thinking in software engineering teaching environments [6] and the complexities associated with setting up a standardised test like the CAT test [51], which requires several days' training.

In this context, only one article [31] proposes a framework close to the evaluation of critical thinking, but not applied specifically to software engineering teaching, suggesting a potential means which should be explored.

C. RQ3: WHAT IS THE PREDOMINANT METHOD OF EVALUATING CRITICAL THINKING IN SOFTWARE ENGINEERING TEACHING?

Fig. 5 shows that 67% of the articles refer to the online mode, while the remaining 33% are based on an offline format. This finding reinforces the hegemonic proposal of remote work aligned with industry, which had already been growing steadily but accelerated during the last year as mentioned by [52].

Similarly, one of the most frequent combinations is "project-based" learning in the online format combined with "teamwork" (see Fig. 10). However the studies reviewed provide few details on the evaluation instruments used, the evaluation strategy or the instruments with which the study was performed.

VI. FRAMEWORK FOR INTEGRATING CRITICAL THINKING INTO VIRTUAL SOFTWARE ENGINEERING TEACHING

Based on analysis of the articles selected, a preliminary framework can be proposed as a guide to setting up an initiative for evaluating critical thinking applied to software engineering teaching in an online context. The proposal is based on four components that interact constantly and repeatedly, and can be combined with the principal active learning strategies detected in the articles (Fig. 14).

Firstly, as proposed in [53], the inclusion of **(a) active, collaborative learning strategies**, such as project-based learning, problem-based learning, case studies and capstone studies, among others, promote a suitable working space for developing an initiative of this type. Selection of one of these strategies will depend on the capacity and experience of the

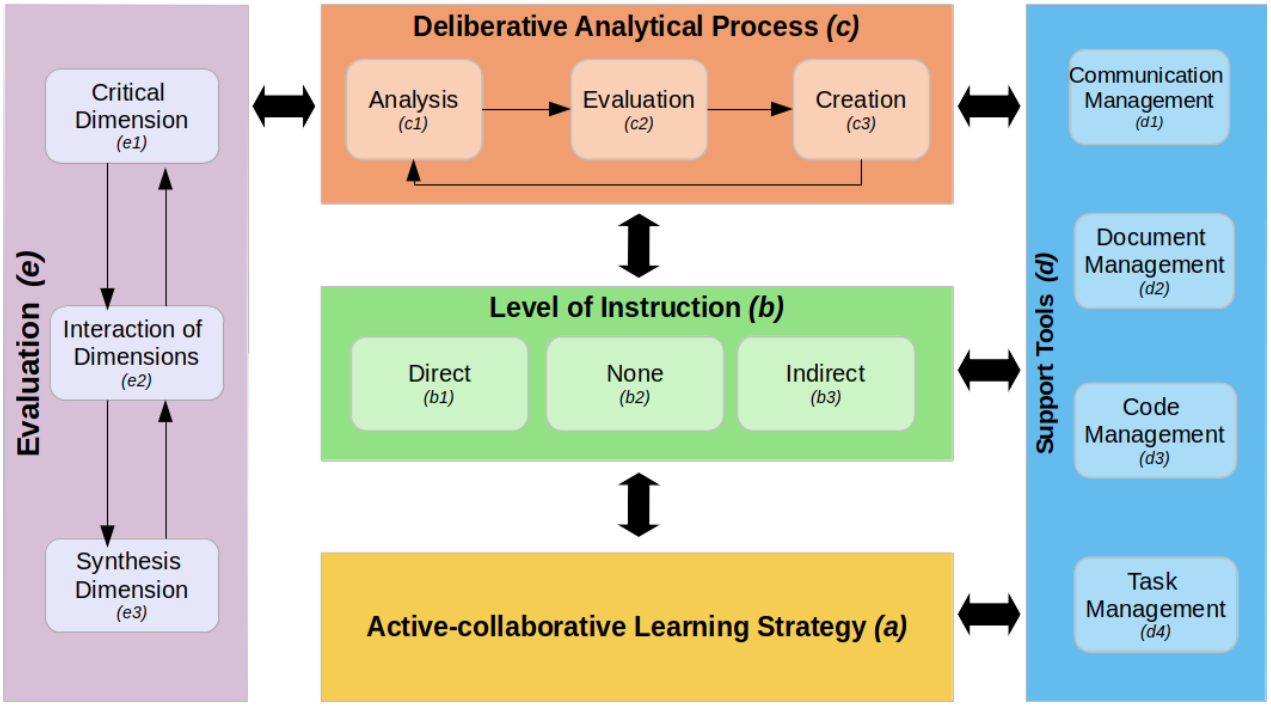

FIGURE 14. Proposed guidelines for the evaluation of critical thinking in the context of software development.

teacher or instructor, as well as the course characteristics. This component is therefore the functional base of the proposal.

Depending on the course characteristics, the second area to address **(b)** is the **level(s) of instruction** covered by the initiative. Direct or indirect instruction, or none, are the categories of participation of the instructor in the course group. These levels must be inclusive and capable of balanced combination, reflecting the different realities and needs of the IT industry.

After these two components, the next **(c)** is the **process of deliberative analysis**, a central element of the reflection process required to solve a particular software problem. Analysis (c1), characterised by argument, questions and exchange of information, seeks to deliberate on the problems inherent in software development in a work environment that is of necessity collaborative. Evaluation (c2) allows the subject to connect and combine elements in search of a solution to a particular problem. Finally, creation (c3) allows all the previous elements to be synthesised into a proposal for a solution. These elements must be iterative, as they must be repeated until individual and group satisfaction is reached among the participants.

The importance of the online mode necessarily requires the incorporation of **support tools (d)** to facilitate synchronic and asynchronic communication (d1); organisation of the documents needed for the work by a documentation manager (d2); management and dissemination of the source code for the whole project (d3); and finally organising and orchestrating the tasks assigned (d4).

The **evaluation** component **(e)** must be adjusted equally to all the other components and avoid the problems associated with generic methods of evaluation, as described in [8]. Thus, following [50], we propose the following dimensions for inclusion in the evaluation of critical thinking in software development environments that are close to industrial:

- **Critical dimension (e1)**: evaluation of the depth of the arguments and the analysis carried out in the Deliberation component, with two central elements: (a) evaluation of evidence collected and used which comes from direct events involving software programming, and b) evaluation of the arguments used from the perspective of guiding a decision which is grounded in the quality of the software and the requirements.
- **Interaction of Dimensions (e2)**: combines the Critical and Synthesis dimensions with the purpose of evaluating whether the student a) understands the causality and the explanation before moving towards a decision. Given the incipient stage of this proposal, the operationalisation of each of the components to be applied has not yet been developed However this area is included in the future research lines of the team, and can be considered as a support for researchers and instructors.
- **Synthesis dimension (e3)**: associated with comprehension of the whole from its parts. It is therefore necessary to evaluate (a) whether the scope of the implications of any decisions taken is understood, and (b) the robustness of the arguments presented.

VII. THREATS TO VALIDITY

This section reports how the biases in the validation of the present work were treated [54].

A. INTERNAL VALIDITY

Threats to internal validity are factors which might affect the results of the present study. The following aspects were considered, with their respective mitigation plans:

- **Search for studies:** To mitigate this threat, we used the pre-defined search chain in the principal electronic databases. Before carrying out a real search in each site, we carried out a pilot search in all the databases selected to verify the accuracy of our search chain.
- **Bias in study selection**: Studies were selected by applying explicitly defined inclusion and exclusion criteria. To avoid possible bias, we also carried out validation by cross-verification for all the studies selected.
- **Bias in data extraction**: To obtain consistent data and avoid biases in data extraction, we defined a summary of the results of the data found. First one of the authors constructed the results distribution tool. Then two authors distributed the number of studies equitably and obtained the data as per the data extraction form. The same two authors discussed their findings and shared them regularly to avoid bias in data extraction.

B. EXTERNAL VALIDITY

Threats to external validity are restrictions which restrict the capacity for generalisation of the results. The inherent threat related with external validity is whether the primary studies report initiatives of critical thinking in software engineering education in the online mode. We mitigated this threat by choosing peer-reviewed studies and excluding grey literature (white papers, editorials, etc.).

C. CONCLUSION VALIDITY

Threats to the validity of the conclusions are issues which affect the ability to draw correct conclusions. Although we used the template of Petersen *et al.* [13], which assumes that it is impossible to identify all the relevant primary studies in existence, we managed this threat to validity by discussing our results in sessions with education professionals and software engineering academics. The number of primary studies obtained for this systematic mapping was such as to enable us to carry out a critical analysis of each one.

D. CONSTRUCT VALIDITY

The validity of the construct is related with the generalisation of the result to the concept or theory behind the execution of the study [54]. The principal threat is subjectivity in our results. To mitigate this threat, all three researchers carried out the main steps of the systematic mapping study independently. Subsequently they discussed their results in order to reach a consensus.

VIII. CONCLUSION

This article presents a preliminary proposal for a framework for the incorporation of critical thinking in the context of software engineering education online. The proposal was developed from a systematic mapping study of articles related with this topic. Three Research Questions were formulated, all of which were analysed using Petersen's approach [13].

The answers to these questions enabled us to detect key elements for evaluating critical thinking in the context of software engineering education, and these were the principal input for drafting our proposal.

Our proposal takes a generic approach, in other words it could be applied in other teaching contexts, not only software engineering. This is the case so long as the object of the topic shares essential features with software development. This is necessary in view of the multidisciplinary nature of this area. Moreover, the need is growing every day due to the way software and the software life cycle are used to solve problems.

It must be stressed that our results are preliminary, with a top-down approach; in future works we will explore operationalisation of the evaluation dimensions. Our work offers a route for research, and support for teachers of software engineering courses.

Future work, within the authors' study framework, will present deeper analysis of the proposal, followed later by a start-up version in an experimental initiative, to obtain indications of how the proposal can be adapted or improved.

APPENDIX A PRIMARY STUDIES

[A1] S. Motogna, A. Marcus, and A.-J. Molnar, "Adapting to online teaching in software engineering courses," in Proceedings of the 2nd ACMSIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, New York, NY, USA, 2020, p.1–6, doi: 10.1145/3412453.3423194.

[A2] A. P. M. Flores and F. M. R. de Alencar, "Competencies development based on thinking-based learning in software engineering: An action-research," in Proceedings of the 34th Brazilian Symposium on Software Engineering, 2020, pp. 680–689.

[A3] D. M. Marutschke, V. V. Kryssanov, and P. Brockmann, "Distributed virtual courses to teach global software engineering: Lessons learned and best practices," in Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning, 2020, pp.256–260.

[A4] C. Günay, A. Doloc-Mihu, R. Barakat, T. Gluick, and C. A.Moore, "Improving Critical Thinking in Software Development via Interdisciplinary Projects at a Most Diverse College," in Proceedings of the 21st Annual Conference on Information Technology Education, New York, NY, USA, Oct. 2020, pp. 206–212, doi: 10.1145/3368308.3415411.

[A5] C. Günay, A. Doloc-Mihu, T. Gluick, and C. A. Moore, "Project-Based Learning Improves Critical Thinking for Software Development Students," in Proceedings of the 20th Annual SIG Conference on Information Technology Education, New York, NY, USA, Sep. 2019, p. 105, doi: 10.1145/3349266.3351362.

[A6] D. Coore and D. Fokum, "Facilitating course assessment with acompetitive programming platform," in Proceedings of the 50th ACM Technical Symposium on Computer Science Education, New York, NY, USA, 2019, p.449–455, doi: 10.1145/3287324.3287511.

[A7] R. Chanin, A. Sales, L. Pompermaier, and R. Prikladnicki, "Challenge based startup learning: A framework to teach software startup," in Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, NewYork, NY, USA, 2018, p. 266–271, doi: 10.1145/3197091.3197122.

[A8] O. Chouseinoglou and S. Bilgen, "Introducing critical thinking to software engineering education," in Software Engineering Research, Management and Applications (Studies in Computational Intelligence), R. Lee, Ed. Berlin, Germany: Springer, 2014, pp.183-195

[A9] C. Álvarez, N. Baloian, G. Zurita, and F. Guarini, "Promoting active learning in large classrooms: Going beyond the clicker," in CYTED-RITOS International Workshop on Groupware. Springer, 2017, pp. 95-103

[A10] J. Cleland-Huang and M. Rahimi, "A Case Study: Injecting Safety-Critical Thinking into Graduate Software Engineering Projects," in 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET), May 2017,pp. 67-76

[A11] B. Kules, "Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes," Proceedings of the association for information science and technology, vol. 53, no. 1, pp. 1–6, 2016

[A12] R. Vivian, K. Falkner, N. Falkner, and H. Tarmazdi, "A method to analyze computer science students teamwork in online collaborative learning environments," ACM Transactions on Computing Education (TOCE),vol.16,no. 2, pp. 1–28, 2016.

[A13] H. Suleman, "Flipping a course on computer architecture," in Annual Conference of the Southern African Computer Lecturers Association. Springer, 2016, pp. 83–94

[A14] J. M. Carroll, H. Jiang, and M. Borge, "Distributed collaborative homework activities in a problem based usability engineering course," Education and Information Technologies, vol. 20, no. 3, pp. 589–617, 2015

[A15] M.A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky, "The (R) evolution of social media in software engineering," in Futureof Software Engineering Proceedings, New York, NY, USA, 2014, p. 100–116, doi: 10.1145/2593882.2593887.

[A16] R. P. D. Redondo, A. F. Vilas, J. J. Pazos Arias, andA. Gil Solla, "Collaborative and roleplay strategies in software engineering learning with web 2.0 tools," Computer Applications in Engineering Education, vol. 22, no. 4, pp. 658–668, 2014, doi: 10.1002/cae.21557.

[A17] L. C. Belcadhi and S. A. Ghannouchi, "An instructional design approach for e-active courses," in Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality, 2013, pp.119–126.

[A18] R. Razali and D.A.P. Zainal, "Assessing students acceptance of case method in software engineering education–a survey," Procedia-Social and Behavioral Sciences, vol.93, pp.1562–1568, 2013, doi: 10.1016/j.sbspro.2013.10.082.

[A19] S. M. Salleh, Z. Tasir, and N. A. Shukor, "Web-based simulation learning framework to enhance students critical thinking skills," Procedia-Social and Behavioral Sciences, vol. 64, pp. 372–381, 2012

[A20] L. Pollock and T. Harvey, "Combining multiple pedagogies to boost learning and enthusiasm," in Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education, New York, NY, USA, 2011, pp. 258–262, doi: 10.1145/1999747.1999820.

[A21] E. C. Pappas, "A behavioral approach to building cognitive foundations for effective thought and action in a freshman critical thinking course," in 2011 Frontiers in Education Conference (FIE). IEEE, 2011, pp. T3C–1

REFERENCES

[1] T. W. E. Forum. (2020). *The Global Risks Report 2019*. World Economic Forum. [Online]. Available: https://www.weforum.org/reports/the-global-risks-report-2019/

[2] Hired. (2020). *2020 State of Software Engineers*. [Online]. Available: https://hired.com/state-of-software-engineers

[3] HackerRank. (2020). *En-USHackerRank Developer Skills Report*. HackerRank. [Online]. Available: https://info.hackerrank.com/rs/487-WAY-049/images/HackerRank-2020-Developer-Skills-Report.pdf

[4] S. Soundararajan, A. Chigani, and J. D. Arthur, "Understanding the tenets of agile software engineering: Lecturing, exploration and critical thinking," in *Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, Feb. 2012, pp. 313–318, doi: 10.1145/2157136.2157228.

[5] A. Ahern, C. Dominguez, C. McNally, J. J. O'Sullivan, and D. Pedrosa, "A literature review of critical thinking in engineering education," *Stud. Higher Educ.*, vol. 44, no. 5, pp. 816–828, May 2019, doi: 10.1080/03075079.2019.1586325.

[6] O. Chouseinoglou and S. Bilgen, "Introducing critical thinking to software engineering education," in *Software Engineering Research, Management and Applications* (Studies in Computational Intelligence), R. Lee, Ed. Heidelberg, Germany: Springer, 2014, pp. 183–195.

[7] M. Exter, S. Caskurlu, and T. Fernandez, "Comparing computing professionals' perceptions of importance of skills and knowledge on the job and coverage in undergraduate experiences," *ACM Trans. Comput. Educ.*, vol. 18, no. 4, pp. 21:1–21:29, Nov. 2018, doi: 10.1145/3218430.

[8] C. Günay, A. Doloc-Mihu, R. Barakat, T. Gluick, and C. A. Moore, "Improving critical thinking in software development via interdisciplinary projects at a most diverse college," in *Proc. 21st Annu. Conf. Inf. Technol. Educ.*, New York, NY, USA, Oct. 2020, pp. 206–212, doi: 10.1145/3368308.3415411.

[9] C. Günay, A. Doloc-Mihu, T. Gluick, and C. A. Moore, "Project-based learning improves critical thinking for software development students," in *Proc. 20th Annu. SIG Conf. Inf. Technol. Educ.*, New York, NY, USA, Sep. 2019, p. 105, doi: 10.1145/3349266.3351362.

[10] J. Cleland-Huang and M. Rahimi, "A case study: Injecting safety-critical thinking into graduate software engineering projects," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng., Softw. Eng. Educ. Training Track (ICSE-SEET)*, May 2017, pp. 67–76.

[11] K. Garg and V. Varma, "A study of the effectiveness of case study approach in software engineering education," in *Proc. 20th Conf. Softw. Eng. Educ. Training (CSEET)*, Jul. 2007, pp. 309–316.

[12] B. Fagin, J. Harper, L. Baird, S. Hadfield, and R. Sward, "Critical thinking and computer science: Implicit and explicit connections," *J. Comput. Sci. Colleges*, vol. 21, no. 4, pp. 171–177, 2006.

[13] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. 12th Int. Conf. Eval. Assessment Softw. Eng.*, Jun. 2008, pp. 68–77.

[14] M. Kennedy, M. B. Fisher, and R. H. Ennis, "Critical thinking: Literature review and needed research," in *Educational Values and Cognitive Instruction*, L. Idol and B. F. Jones, Eds. New York, NY, USA: Lawrence Albourne Associated, 1991, ch. 1, pp. 11–35.

[15] R. J. Sternberg, "Critical thinking: Its nature, measurement, and improvement," Nat. Inst. Educ. (ED), Washington, DC, USA, Tech. Rep., 1986. [Online]. Available: https://eric.ed.gov/?id=ED272882

[16] E. R. Lai, "Critical thinking: A literature review," *Pearson's Res. Rep.*, vol. 6, no. 1, pp. 40–41, 2011.

[17] P. A. Facione, "The disposition toward critical thinking: Its character, measurement, and relationship to critical thinking skill," *Informal Log.*, vol. 20, no. 1, pp. 61–84, Jan. 2000.

[18] D. Hitchcock, "Critical thinking," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed. Stanford, CA, USA: Stanford Univ., 2020.

[19] D. T. Willingham, "Critical thinking: Why is it so hard to teach?" *Arts Educ. Policy Rev.*, vol. 109, no. 4, pp. 21–32, Mar. 2008.

[20] R. H. Ennis, "Critical thinking and subject specificity: Clarification and needed research," *Educ. Res.*, vol. 18, no. 3, pp. 4–10, Apr. 1989.

[21] E. Silva, "Measuring skills for 21st-century learning," *Phi Delta Kappan*, vol. 90, no. 9, pp. 630–634, May 2009, doi: 10.1177/003172170909000905.

[22] T. V. Gelder, "Teaching critical thinking: Some lessons from cognitive science," *College Teaching*, vol. 53, no. 1, pp. 41–48, Jan. 2005, doi: 10.3200/CTCH.53.1.41-48.

[23] R. T. Pithers and R. Soden, "Critical thinking in education: A review," *Educ. Res.*, vol. 42, no. 3, pp. 237–249, 2000.

[24] K. Ulger, "The effect of problem-based learning on the creative thinking and critical thinking disposition of students in visual arts education," *Interdiscipl. J. Problem-Based Learn.*, vol. 12, no. 1, p. 10, Mar. 2018.

[25] H. Mora, M. T. Signes-Pont, A. Fuster-Guilló, and M. L. Pertegal-Felices, "A collaborative working model for enhancing the learning process of science & engineering students," *Comput. Hum. Behav.*, vol. 103, pp. 140–150, Feb. 2020.

[26] M. L. Styers, P. A. Van Zandt, and K. L. Hayden, "Active learning in flipped life science courses promotes development of critical thinking skills," *CBE Life Sci. Educ.*, vol. 17, no. 3, p. ar39, Sep. 2018.

[27] S. F. D. Vargas, M. F. A. Fajardo, E. T. A. Cedillo, and C. Electrónicos, "La importancia del protagonismo estudiantil: Aprendizaje activo y cooperativo," in *Memorias del Quinto Congreso Internacional de Ciencias Pedagógicas de Ecuador: Aprendizaje en la Sociedad del Conocimiento: Modelos, Experiencias y Propuestos*. Guayaquil, Ecuador: Instituto Superior Tecnológico Bolivaria, 2019, pp. 1049–1055.

[28] C. Cortázar, M. Nussbaum, J. Harcha, D. Alvares, F. López, J. Goñi, and V. Cabezas, "Promoting critical thinking in an online, project-based course," *Comput. Hum. Behav.*, vol. 119, Jun. 2021, Art. no. 106705.

[29] M. Pedaste, M. Mäeots, L. A. Siiman, T. de Jong, S. A. N. van Riesen, E. T. Kamp, C. C. Manoli, Z. C. Zacharia, and E. Tsourlidaki, "Phases of inquiry-based learning: Definitions and the inquiry cycle," *Educ. Res. Rev.*, vol. 14, pp. 47–61, Feb. 2015.
[30] J. Valenzuela and A. M. Nieto, "Motivación y pensamiento crítico: Aportes para el estudio de esta relación," *REME*, vol. 11, no. 28, 2008. [Online]. Available: https://www.academia.edu/download/49226601/article3.pdf
[31] S. M. Salleh, Z. Tasir, and N. A. Shukor, "Web-based simulation learning framework to enhance Students' critical thinking skills," *Proc. Social Behav. Sci.*, vol. 64, pp. 372–381, Nov. 2012.
[32] L. C. Belcadhi and S. A. Ghannouchi, "An instructional design approach for e-active courses," in *Proc. 1st Int. Conf. Technol. Ecosyst. Enhancing Multiculturality*, 2013, pp. 119–126.
[33] D. M. Marutschke, V. V. Kryssanov, and P. Brockmann, "Distributed virtual courses to teach global software engineering: Lessons learned and best practices," in *Proc. 11th Int. Conf. E-Educ., E-Bus., E-Manage., E-Learn.*, Jan. 2020, pp. 256–260.
[34] B. Kules, "Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes," *Proc. Assoc. Inf. Sci. Technol.*, vol. 53, no. 1, pp. 1–6, 2016.
[35] K. Y. L. Ku, "Assessing students' critical thinking performance: Urging for measurements using multi-response format," *Thinking Skills Creativity*, vol. 4, no. 1, pp. 70–76, Apr. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1871187109000054
[36] S. C. Fischer, V. A. Spiker, and S. L. Riedel, "Critical thinking training for army officers. Volume 2: A model of critical thinking," U.S. Army Res. Inst. the Behav. Social Sci., Arlington, VA, USA, Tech. Rep. 1882, 2009.
[37] B. Stein, A. Haynes, M. Redding, T. Ennis, and M. Cecil, "Assessing critical thinking in stem and beyond," in *Innovations in E-learning, Instruction Technology, Assessment, and Engineering Education*, M. Iskander, Ed. Dordrecht, The Netherlands: Springer, 2007, pp. 79–82.
[38] S. M. Glynn and K. D. Muth, "Reading and writing to learn science: Achieving scientific literacy," *J. Res. Sci. Teaching*, vol. 31, no. 9, pp. 1057–1073, Nov. 1994.
[39] C. I. de Desarrollo-Cinda and G. O. D. U. Chilenas, "Evaluación del aprendizaje en innovaciones curriculares de la educación superior," CINDA Centro Interuniversitario de Desarrollo, Santiago, Chile, Tech. Rep., 2014. [Online]. Available: https://cinda.cl/publicacion/evaluacion-del-aprendizaje-en-innovaciones-curriculares-de-la-educacion-superior/
[40] H. Hashim, M. N. Ali, and M. A. Shamsudin, "Infusing high order thinking skills (HOTs) through thinking based learning (TBL) during ECA to enhance students interest in STEM," *Int. J. Academic Res. Bus. Social Sci.*, vol. 7, no. 11, pp. 1191–1199, Dec. 2017.
[41] A. P. M. Flores and F. M. R. de Alencar, "Competencies development based on thinking-based learning in software engineering: An action-research," in *Proc. 34th Brazilian Symp. Softw. Eng.*, Oct. 2020, pp. 680–689.
[42] K. Gwet, "Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity," *Stat. Methods Inter-Rater Rel. Assessment*, vol. 2, p. 9, Jan. 2002.
[43] R. Chanin, A. Sales, L. Pompermaier, and R. Prikladnicki, "A systematic mapping study on software startups education," in *Proc. 22nd Int. Conf. Eval. Assessment Softw. Eng.*, Jun. 2018, pp. 163–168.
[44] N. L. Veras, L. S. Rocha, and W. Viana, "Flipped classroom in software engineering: A systematic mapping study," in *Proc. 34th Brazilian Symp. Softw. Eng.*, Oct. 2020, pp. 720–729.
[45] V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer, "Aligning software engineering education with industrial needs: A meta-analysis," *J. Syst. Softw.*, vol. 156, pp. 65–83, Oct. 2019.
[46] L. Anthonysamy, A.-C. Koo, and S.-H. Hew, "Self-regulated learning strategies and non-academic outcomes in higher education blended learning environments: A one decade review," *Educ. Inf. Technol.*, vol. 2020, pp. 1–28, Feb. 2020.
[47] H. Suleman, "Flipping a course on computer architecture," in *Proc. Annu. Conf. Southern Afr. Comput. Lecturers' Assoc.* South Africa: Springer, 2016, pp. 83–94.
[48] G. Mason, T. R. Shuman, and K. E. Cook, "Inverting (flipping) classrooms-advantages and challenges," in *Proc. 120th ASEE Annu. Conf. Exhib.*, Atlanta, GA, USA, Jun. 2013, pp. 23–828. [Online]. Available: https://www.asee.org/public/conferences/20/papers/7171/download
[49] M. R. Marques, A. Quispe, and S. F. Ochoa, "A systematic mapping study on practical approaches to teaching software engineering," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2014, pp. 1–8.
[50] O. L. Liu, L. Frankel, and K. C. Roohr, "Assessing critical thinking in higher education: Current state and directions for next-generation assessment," *ETS Res. Rep.*, vol. 2014, no. 1, pp. 1–23, Jun. 2014, doi: 10.1002/ets2.12009.
[51] T. T. University. *Center for Assessment and Improvement of Learning*. Accessed: Apr. 1, 2021. [Online]. Available: https://www.tntech.edu/cat/
[52] M. Wohlmuth. (2021). *En-USState of Software Development 2021*. Codingsans. [Online]. Available: https://codingsans.com/state-of-software-development-2021
[53] N. Giacaman and O. Sinnen, "Preparing the software engineer for a modern multi-core world," *J. Parallel Distrib. Comput.*, vol. 118, pp. 247–263, Aug. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731518301060
[54] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012.

OSCAR ANCÁN BASTÍAS received the B.Eng. and M.S. degrees in informatics from the Universidad de La Frontera, Temuco, Chile, in 2008 and 2018, respectively. He is currently a Professor with the Department of Computer Science and Informatics, Universidad de La Frontera. His research interests include software maintenance, software architecture, smart cities architecture, and software engineering education.

JAIME DÍAZ received the M.Sc. and Ph.D. degrees in engineering informatics. He is a Technological Projects Evaluator at CORFO (a Chilean Government Organization) and a UX Researcher. He is currently working as a full-time Professor with the Universidad de la Frontera, Chile. His research interests include HCI, eCommerce, and education.

CRISTIAN OLIVARES RODRÍGUEZ was a Computer Scientist at the Catholic University of Concepción, Concepción, Chile, in 2006. He received the M.S. degree in computer vision and artificial intelligence from the Autonomous University of Barcelona, Barcelona, Spain, in 2010, and the Ph.D. degree in engineering from the University of Deusto. He is interested in modeling human behavior by means of data science, with applications in education and heuristic problems.