ARTICLE IN PRESS

The Journal of Systems & Software xxx (xxxx) xxx

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends

Orges Cico [a,*], Letizia Jaccheri [a], Anh Nguyen-Duc [b], He Zhang [c]

[a] Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway
[b] Department of Business and IT, USN School of Business, Bø i Telemark, Norway
[c] DevOps+ Research Laboratory, Nanjing University, China

ARTICLE INFO

Article history:
Received 14 October 2019
Received in revised form 11 May 2020
Accepted 9 July 2020
Available online xxxx

Keywords:
Software industry
Software Engineering Education
Software Engineering Trends
Industry education intersection
Systematic mapping study

ABSTRACT

Context: Software has become ubiquitous in every corner of modern societies. During the last five decades, software engineering has also changed significantly to advance the development of various types and scales of software products. In this context, Software Engineering Education plays an important role in keeping students updated with software technologies, processes, and practices that are popular in industries.
Objective: We investigate from literature the extent Software Engineering Education addresses major Software Engineering Trends in the academic setting.
Method: We conducted a systematic mapping study about teaching major Software Engineering Trends in project courses. We classified 126 papers based on their investigated Software Engineering Trends, specifically Software Engineering processes and practices, teaching approaches, and the evolution of Software Engineering Trends over time.
Results: We reveal that Agile Software Development is the major trend. The other Trends, i.e., Software Implementation, Usability and Value, Global Software Engineering, and Lean Software Startup, are relatively small in the academic setting, but continuously growing in the last five years. System of Systems is the least investigated among all Trends.
Conclusions: The study points out the possible gaps between Software Industry and Education, which implies actionable insights for researchers, educators, and practitioners.

1. Introduction

As in the last five decades, Software Engineering Education (SEE) continues to evolve, with the main focus being the preparation of Software Engineering (SE) students for future careers (Marques et al., 2014a; Almi et al., 2011). International organizations, such as the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE) (A. f. C. M. A. Joint Task Force on Computing Curricula, I. C. Society, 2013), and Computing Curricula of 2020 (Impagliazzo et al., 2018), guide SE curricula with consideration of integrating industrial perspectives. Nevertheless, addressing industrial demands is still an open question for SEE.

Educators provide fundamental programming knowledge and skills which help students work with new technologies in industrial environments. In ongoing efforts, SEE strives to meet this goal by designing courses that have a longer duration and rely on different teaching strategies (e.g., project-based, problem-based, studio-based learning) to enable students to practice their skills in relatively realistic project environments (Beckman et al., 1997; Jaakkola et al., 2006; Almi et al., 2011; Kuang and Han, 2012; Bull et al., 2013; Loksa et al., 2013). Previous systematic reviews (Marques et al., 2014b; Garousi et al., 2016; Beecham et al., 2017b) reveal that educators manage to teach relevant Software Engineering knowledge to students.

Moreover, education for software engineers should prepare students to stay current in the face of rapid change. Existing studies report educational challenges that exceed fundamental skill sets. For example, reports exist that address how to support students to communicate effectively with customers in an Agile project (Lethbridge et al., 2007) and how to work with other developers in a geographically distributed setting. The underlying assumption of these reports is that students already know the required state-of-the-art SE Trends. Relevant topics are essential for educators in building an appropriate curriculum and selecting

* Correspondence to: IT-bygget, 102, Gløshaugen, Sæm Sælandsvei 7, Trondheim, Norway.
E-mail address: orges.cico@ntnu.no (O. Cico).

https://doi.org/10.1016/j.jss.2020.110736
0164-1212/

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

a suitable teaching methodology (Bass, 2016; Bolinger et al., 2010).

Triggered by the vision of preparing students for future computing from the Computing Curricula 2020 (CC2020) project (Clear et al., 2019), we recognize the necessity for reviewing SEE in order to yield educational outcomes relevant to the Software industry in the mid-2020s and beyond. Previous studies highlight the importance for SEE actors to collaborate on common education goals, as well as remain current with SE Trends (Beckman et al., 1997; Garousi et al., 2016). In particular, the SE Trends from previous decades that are within SE curriculum need revision. In this study, SE Trend is defined as a commonly adopted software development paradigm, which includes SE ideology, methodology, way of working, a framework, a process, and a set of practices. For example, at present, DevOps is a popular SE Trend gaining widespread adoption in the software industry (Bezemer et al., 2019).

Decisions to update curricula for SEE teaching needs (i.e., current trends in the SE industry is reflected in the academic setting) (Boehm, 2006) must be informed by in depth understanding of state-of-the-art SE Trends. In order to provide an overview of the investigated area, we conducted a systematic mapping study. We classified 126 papers selected for review into the following categories: the SE Trends, teaching strategy, models/processes, methods, participating stakeholders, research, and contribution type. Additionally, we assessed publication trends and sources to understand the evolution and quality of the conducted research. The contribution of the work includes:

- Scoping of the research area in SEE and identifying the endorsed SE Trend
- Identifying the quantity and type of research available
- Mapping the frequencies of publications over time
- Assessing the current teaching strategies reflected in SEE research as scientific publications
- Evaluating stakeholder joint efforts for participation in industry and SEE
- Identifying research gaps in SEE for future study

We found that 79.4% of SEE studies are associated with the most common SE Trend, namely Agile Software Development. Software Implementation, Usability and Value is the second most explored SE Trend found in 16.7% of SEE studies. Other SE Trends, such as Lean Software Startup, Global Software Engineering (GSE), are shown in only c.a 9.5% of SEE studies. The SE Trend System of System is little explored in SEE context.

Concerning learning approaches, Project-based learning approaches are common in SEE Other potential learning approaches, i.e., gamified learning and blended learning, are still little explored. The most common methodological approach in SEE context is Scrum. There are reports, however, about the combination of Agile and Lean approaches in education for Startups. Finally, the actual participation of Industrial stakeholders in SEE is still limited.

The study structure is as follows. In Section 2, we provide the research background. Section 3 contains explanation for the systematic mapping approach used for selecting papers. In Section 4, we analyze the results and answer the research questions (RQs). Section 5 includes the discussion on the findings and limitations of the research. Then, in Section 6, we conclude and propose future research work. The complete information and classification of the mapping can be found in Appendix.

2. Research background

In this study, a SE Trend is defined as a commonly adopted software development paradigm, which includes SE ideology, methodology, way of working, a framework, a process, and a set of practices. This is conceptualized into a search for software models, processes, methods, and practices in our search protocol. We later use the same scope and granularity level when addressing RQs of our study.

Shaw, in her paper on SEE, in 2000, stated the relevance of SE Trends to education, *"Changes in software technology and models for software development require commensurate change in the education of software developers"* (Shaw, 2000). Based on Shaw's recommendation, we looked into (1) literature predicting future SE Trends, and (2) existing systematic reviews in SEE research.

2.1. Software industry trends

Soon after Shaw, in 2006, Boehm provides an overview of the SE perspective during the 20th and 21st-century. Boehm constructs a comprehensive timeline of the SE Trends since the 1950s up to 2025. The author identifies 10 future SE Trends: (1) Rapid change and the need for agility, (2) Increased emphasis on usability and value, (3) Software criticality and the need for dependability, (4) Increasing needs for COTS, reuse, and legacy software integration, (5) Increasing integration of software and systems engineering, (6) Global connectivity, (7) Massive System of Systems, (8) Computational plenty, (9) Increasing software autonomy, and (10) Combinations of biology and computing. Boehm predicts that development paradigms (Agile and Value-based SE) regarding Startups and Globalization are expected to be extensively present in SE practice after the 2000s and 2010s. Important topics related to GSE (Monasor et al., 2010; Clear et al., 2016; Clear and Beecham, 2019) and Continuous Software Development (Zhang et al., 2010; Krusche and Alperowitz, 2014a) are also stated in academic contexts. Boehm's study covers a comprehensive list of SE trends. In the duration between 2000 and 2019, we performed a manual search on recognized SEE journals and conferences, such as Journal of Systems and Software, Information Technology Journal, Information, and Software Technology, ACM Transactions on Computing Education, International Conference on Software Engineering, Frontiers in Education, Conference on Software Engineering Education and Training, and International Conference on Global Software Engineering. Searched terms were input into search engines are SE trends and its synonyms, i.e., software trends, software engineering themes, and software engineering future directions. Among the results, we consulted research experts about SE and selected the papers with high citations.

We find that Fitzgerald and Stol (2017) identifies four notable areas as part of the recent software engineering trends: (1) Enterprise Agile, (2) DevOps, (3) Beyond budgeting, and (4) Lean Software Startups. Zhang et al. (2010) provide a similar overview, with emphasis on (1) Agile, (2) Lean Development, and (3) Continuous Testing. Dingsøyr et al. (2012) have also made an extensive evaluation of Agile practices in the industry.

Other mentioned upcoming SE Trends are expected to relate closely to hardware technological advancements (sensor networks, conformable or adaptive materials, human prosthetics) and exploit new technological materials (smart materials, nanotechnology, micro-electrical mechanical systems) (Boehm, 2006). Software engineering at System of Systems levels covers the development and maintenance of functional and non-functional attributes at a higher and complex setting. Further areas of investigation involve software autonomy, as well as combining biology and computing.

Aggregated from the above references, we devise a list of the SE Trends found in literature from the last two decades, as shown in Table 1. We use the list of SE Trends later to identify the intersection with education SE Trends, reporting the common SE Trends within the industry-education areas.

Table 1
Common modern SE Trends according to SE educational references.

Trend start year	Software engineering trends	References
1970's/1980's/1990's	Reusability, COTS, Open Source Software	Boehm (2006) and Shaw (2000)
1990's	Usability, User Experience, HCI	Boehm (2006)
2000's	Agility, Agile Software Development	Boehm (2006) and Zhang et al. (2010)
2000's	Dependability, i.e. Safety, Security and Trust	Boehm (2006) and Shaw (2000)
2005	Enterprise Agile, Large-scale Agile	Fitzgerald and Stol (2017)
2010's	Global Software Engineering	Boehm (2006) and Monasor et al. (2010), and Clear et al. (2016), Clear and Beecham (2019)
2010's	Lean Startup Education	Järvi et al. (2015), Devadiga (2017) and Fitzgerald and Stol (2017)
2010's	Continuous Software Engineering	Krusche and Alperowitz (2014b) and Zhang et al. (2010)
2010's	System of system engineering	Boehm (2006) and Shaw (2000)
2012	DevOps	Fitzgerald and Stol (2017)
2020's	Computational plenty	Boehm (2006)
2020's	Software Engineering for autonomous systems	Boehm (2006)
2025	Biology and computing	Boehm (2006)

2.2. Existing systematic reviews in software engineering education

The software engineering discipline has evolved throughout the past 50 years, guided by the Software Engineering Body of Knowledge (SWEBOK). Software engineering students usually focus primarily on computer science. Their careers orient them toward well-established and large companies, wherein training is typical for providing software engineering skills (Programming languages, frameworks and tools, technologies) to lower-level employees. Decision making is within the scope of senior developers' or project managers. The lack of decision making leaves few opportunities for recently employed students to invest in soft skills (resume building, career planning, communication, teamwork and collaboration, time management, presentation planning, and managing with learning challenges) (Begel and Simon, 2008a,b; Pulko and Parikh, 2003; Carter, 2011). However, today's expectations are for students to be better problem solvers and to make rapid decisions during face-to-face communication with end customers, including in the SEE context (Rico and Sayani, 2009; Marques et al., 2018; Paasivaara et al., 2018). Involving external industry stakeholders, as discussed in Steghöfer et al. (2018), has played a vital role in students' technical and soft skills.

Software Engineering Education has continuously struggled to provide SE students with appropriate skills to excel in their jobs. Marques and Garousi (Marques et al., 2014b; Garousi et al., 2016) made previous efforts to map studies related to teaching practices in software engineering. Other recent publications have explored GSE as part of software engineering teaching strategies (Clear et al., 2015; Beecham et al., 2017b). However, the need for exploring which industry trends are encompassed in those strategies and those needing further exploration are becoming significant for informing future decisions. From previous literature reviews (Clear et al., 2015; Beecham et al., 2017b), we determine to focus on project courses adopting realistic industry practices while addressing a particular trend (e.g., GSE).

3. Research methodology

3.1. Motivation for conducting the systematic literature mapping

The study objective is to identify to what extent SE Trends are present in SEE research. Additionally, we characterize the associated teaching approaches and involvement of relevant stakeholders in the reported project courses. We present the scope of the study in Fig. 1.

The original focus is on the intersection between the current state of SEE (designated in blue) and SE Trends (designated in yellow). From the researchers' perspective, we partially explore this intersection by looking at how SEE adopts SE Trends appearing in industry. To achieve a complete overview of the area, we comprehensively review SEE papers that address SE Trends identified in Section 2.1.

Previous efforts mapped studies related to teaching software engineering (Marques et al., 2014b), while more recent publications explored challenges, best practices (Garousi et al., 2016), and GSE (Beecham et al., 2017b) in SEE. However, to our best knowledge, within the last decade, there is not a recent effort to systematically map common SE Trends between industry and education. Our mapping potentially allows researchers and practitioners to make informed decisions regarding SEE, while reflecting present and future industry needs.

3.2. Systematic mapping study

Petersen et al. (2008) suggest that, by categorizing the papers, a systematic mapping study provides a structure for the type of research reports and results that have been published. The first step of the process involves posing RQs, which then help to generate a visual summary of the research results. The other steps involve screening based on title, abstract, and keyword metadata. The results help answer the RQs. The primary focus of a systematic mapping study is to identify gaps in the research area under investigation. Fig. 2 represents the systematic mapping process (Petersen et al., 2008) we have followed.

3.2.1. Definition of RQs

Deriving from the study motivation and objective (Section 3.1), we propose four primary RQs:

- **RQ1:** To what extent are **SE Trends** presented in **SEE research**?
  - **RQ1.1:** What is the distribution of SE Trends in **SEE research**?
  - **RQ1.2:** Which of the industry **models, processes, and methods** are embraced in **SEE research**?
  - **RQ1.3:** How have the SE Trends in **SEE research** evolved over time?
- **RQ2:** How does **SEE research** present the teaching of various **SE Trends**?
  - **RQ2.1:** Which are the **industry-relevant teaching** approaches presented in **SEE research**?
  - **RQ2.2:** Which **stakeholders** worked together as presented from **SEE research**?
- **RQ3:** How do **SE Trends** contribute to literature?
- **RQ4:** Which bibliographical sources primarily publish studies?

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

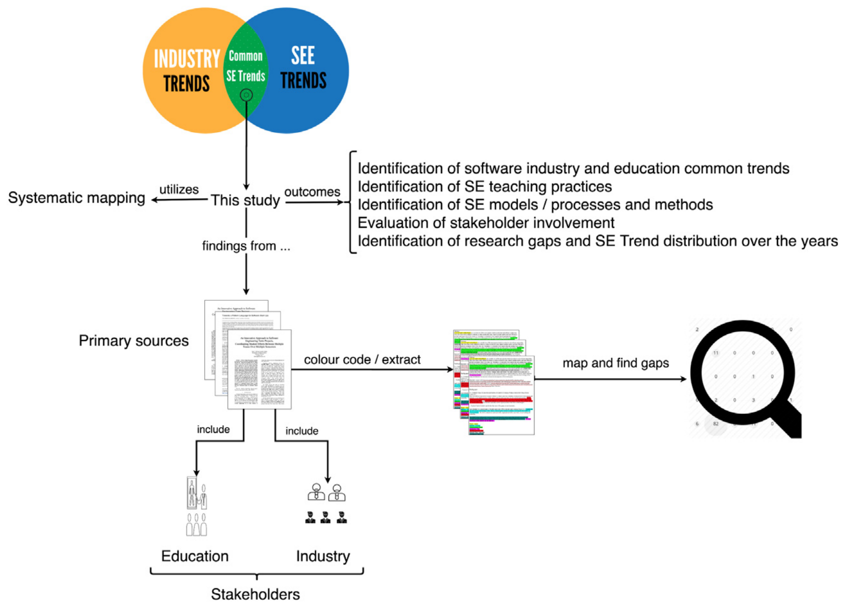


Fig. 1. Research context utilized in conducting the mapping study. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

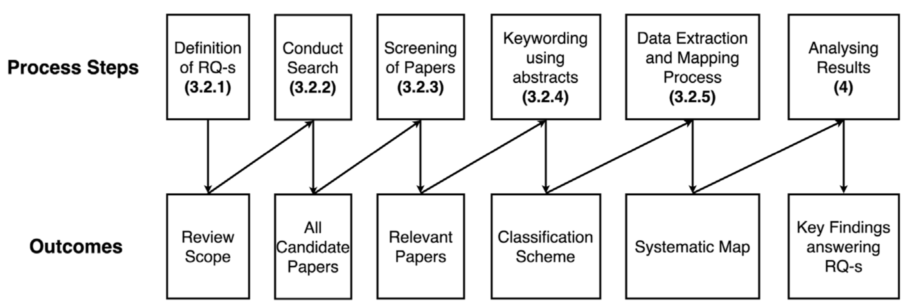


Fig. 2. The systematic mapping process (Petersen et al., 2008).

Table 2
PICO of the study.

Population	Primary studies in SEE (both theoretical and empirical studies)
Intervention	Academic settings, i.e., teaching approaches, models, methods, frameworks, and stakeholders
Comparison	Technologies, methods, frameworks, stakeholders in SEE projects, and industrial projects
Outcome	Evaluation of empirical studies collected from SEE and their adherence to the industry trends

3.2.2. Conduct search

According to the systematic mapping process, conducting the search involves first identifying the search string. Population, intervention, comparison, and outcome (PICO) criteria, according to Kitchenham and Charters (2007), are defined in Table 2.

To reveal SE Trends occurring in primary studies, we formulated search strings that reveal industrial perspectives on SE courses. The information about SE Trends, such as characteristics of trends (RQ1), teaching approaches (RQ2), research contribution type (RQ3), and publication channels (RQ4) will be extracted from the papers. The details of this keywording process are described in Section 3.2.4.

There are two main elements of the search string "Software Industry" and "Software Engineering Education". Initial searches utilized, at most, two key terms from the first RQ. Afterwards, we augmented the string by including more terms and their synonyms to obtain a more extensive scope of the search: **Basic string**: *(software industry AND software engineering education)* **Full string**: *("software industry" OR "industry trend" OR "industrial client" OR "industrial customer") AND ("software engineering") AND (education or teaching or learning or course)*

We used terms such as "industrial client and customer" to cover SE courses with industrial customer involvement. It is fair to assume that these courses are influenced by industrial topics, themes, knowledge relating to SE process, practices, and methodologies. Many SEE papers might have a primary goal to close the gap between education and practice while introducing a particular software industry trend. We decided that terms such as "software industry" and "industry trend" interchangeable in this work to identify studies that (1) address a particular SE Trend as the main investigated objective and (2) involve industrial customers in academic activities of the courses. We added the remaining terms to identify studies that cover the SEE context.

An investigation of various systematic literature reviews in SE showed several options for electronic databases, such as Scopus, ISI Web of Science, IEEE Xplore, Computer Database, Science Direct, Springer Link, Inspec, and ACM Digital Library. According to our research objective, the selected databases must cover both literature about SE and Education research. We decided to select a set of index databases that complement one another, provide good coverage, and are easy-to-use, as shown in Table 3.

To correctly manage references for removing duplicates and storing a large number of findings, we used the BibDesk reference

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 3
Main sources utilized for the mapping study.

Source type	Denomination
Digital libraries	IEEE Xplore, ACM Digital Library
Databases	Scopus, DBLP bibliography

Table 4
Number of studies per database and Search strings (Time interval : 2008 - 2018)

Source	Basic string	Full string	Library total
ACM	442	40	482
IEEE Xplore	1257	128	1385
Scopus	53	1	54
DBLP	9	0	9
String total	1761	169	**Overall total: 1930**

Table 5
Conference venue manual search.

Venue	Venue total
International Conference on SE (ICSE)	6
International Conference on Software Engineering Education and Training (CSEE&T)	12
Frontiers in Education Conference (FIE)	6
Global Engineering Education Conference (EDUCON)	5
Manual search total	29

manager. The search process took place from the beginning of January 2019 until end of March 2019. The search strings were applied to the meta-data (title, abstract, keywords) from all the sources in Table 3. The **Basic and Full string** are adapted to fit the search database. We restricted publications to those published from 2008 to 2018. To decide on the upper bound of our search year (2018) we are guided by our RQs, recommendations from Zhang et al. (2011), and other Systematic studies (Marques et al., 2014b; Garousi et al., 2016; Beecham et al., 2017b).

Our study sparkles from SE Trends proposed by Boehm in 2006. To define the lower bound of our search year (2008) we checked when (Boehm, 2006) started getting cited in SEE context. The search results per database are in Table 4. To decrease the risk of missing relevant studies, we followed guidelines from Zhang et al. (2011). To this end, apart from the formal database search, we also performed a manual on relevant venues, Table 5.

3.2.3. Screening of papers

The screening process identified the most relevant papers based on the research questions and this mapping study. For each study found with the search string, we decided whether to include it by considering the title, abstract, and keywords. The first and second author assessed the papers during the initial automatic database search, keeping in consideration the research quality, and relation to the RQs. From the database search, the authors identified a total of 1930 papers published during 2008–2018 time span. Most of the publications are in the ACM/IEEE Xplore digital libraries.

Phase 1 involved the automatic removal of 41 invalid sources not meant for citation, such as conference/workshop programs, keynotes, book covers, and unpublished works. Furthermore, with the help of BibDesk, authors automatically removed 112 duplicate papers. Thus, 1777 references remained. In phase 2, authors applied further filtering, based on the inclusion and exclusion criteria, Table 6, first to titles and then to abstracts and occasionally to full text, producing 248 and 97 papers, respectively. We note that, many papers identified as noise (1529) and, thus, filtered out by the first author based on title only as recommended from Petticrew and Roberts (2008).

The search process continued by the first author, with manual searching, adding 29 more papers, Phase 3. Manually found papers are reviewed for inclusion together with the third author. When in doubt, all the first three authors discussed the paper and decided on the outcome. Whenever classification discussions arose, and mutual agreement was not reached, the fourth author, guided the decision. The authors participation, both separately and jointly, during the review process aided in mitigating any bias and threats to the research validity discussed in Section 5.6. We used a collaborative spreadsheet to facilitate the collaboration. The number of included and excluded papers for each Phase is presented in Fig. 3.

3.2.4. Keywording using abstracts and classification scheme generation

The goal of keywording is to create efficiently a classification schema, ensuring that all relevant papers receive consideration. To classify the papers, we followed the process documented in Fig. 4, as proposed by Petersen et al. (2008), who rated classification schemes based on a set of quality attributes:

- Scheme definition - We should define the scheme based on existing literature. To achieve this, we undergo an exhaustive analysis of research in the field of SE Trends in SEE context while determining the taxonomy/classification of the papers.
- Scheme terminology - We should label categories in the scheme while applying terms in the existing literature related to SEE.
- Orthogonality - We should build clear boundaries among the scheme categories, thus making selected relevant papers in SEE easy to classify.
- Completeness - We should make sure to cover all categories so that we can fully classify relevant papers from the SEE context.
- Acceptance - The community accepts and recognizes the classification/taxonomy we use for our study.

The keywording phase consists of the following three steps:

- Step 1: Reading the abstracts of the primary studies and assigning them a set of keywords to identify the main contribution area of the relevant paper. We use cluster categorization based on keywords, as suggested by Petersen et al. (2008), similar to open coding of grounded theory (Charmaz and Belgrave, 2007). We also make sure that keywords are strictly related to the RQs. In this way, we can later classify and map papers accordingly.
- Step 2: Organizing the keywords into a set of categories, each representing the research area of the relevant primary studies. We progressively included the papers into categories, which are, in turn, refined and updated while accommodating new data.
- Step 3: Classifying categories from Step 2 into higher hierarchical levels, recognized as facets, Fig. 5.

One example from two papers we extracted keywords to generate a category within a facet is provided in Fig. 6.

We can observe that keywords derived from the abstracts' and the papers' full-text match, and thus we group the papers in one particular category, which in turn we arrange into facets. Since the process is exhaustive and time-consuming, we relied on the BibDesk reference manager to handle the comprehensive set of papers.

The research context is the basis for the first facet, from which we can identify common SE Trends within industry and SEE **(RQ1.1)**. The second facet concerns the models/processes and

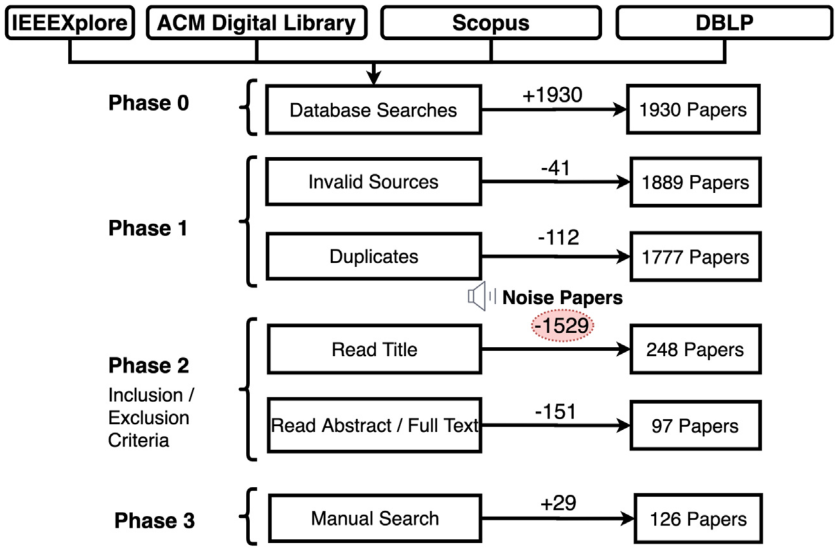


Fig. 3. Selection of primary studies (Petersen et al., 2008).

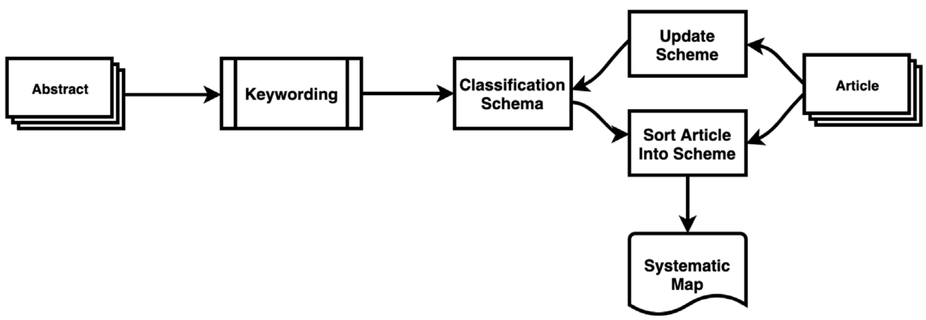


Fig. 4. Classification scheme process (Petersen et al., 2008).

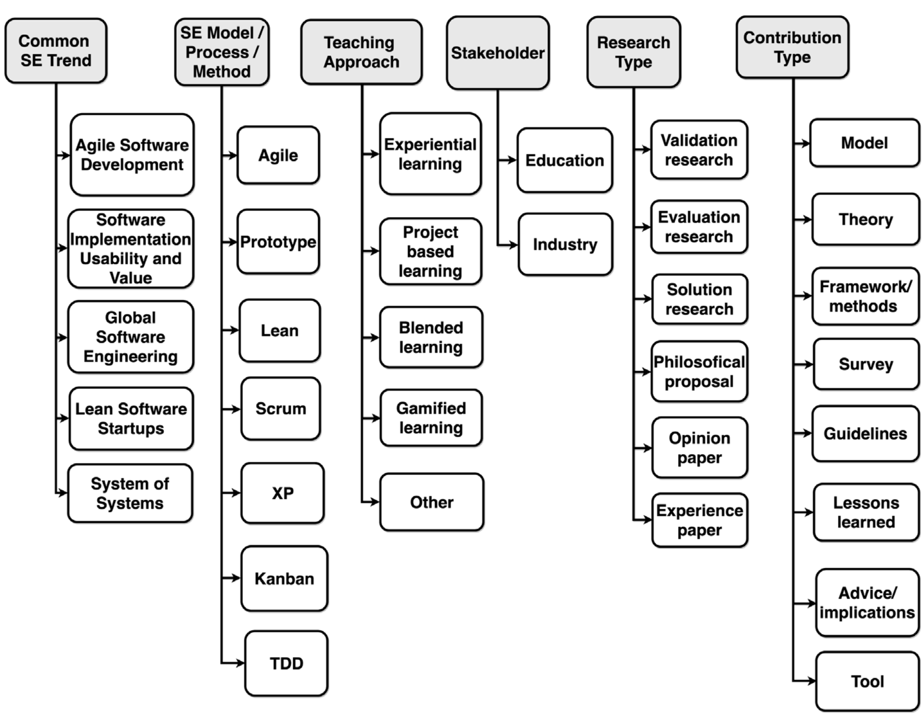


Fig. 5. Classification scheme derived from keywording using abstracts.

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 6
Criteria used for including and excluding studies.

Inclusion criteria	Exclusion criteria
– Included content on SEE that investigates industrial perspectives, i.e., industrial customers, frameworks, and cases	– Mention of software engineering is tangential with different scopes not directly related to industrial aspects
– Written in English with full-text available	– Focus not on software engineering but about other engineering contexts
– Reported on academic settings – Published between 2008 – 2018	– Presentation is of non-peer reviewed material – Presentation is not in English – Full-text is inaccessible – Books and gray literature – Studies that duplicate other studies

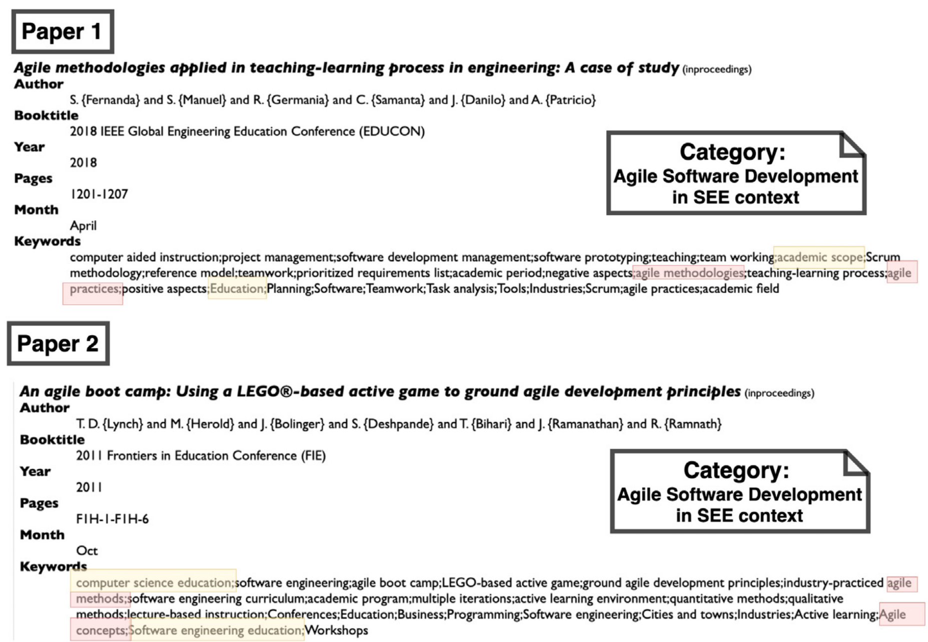


Fig. 6. Example category creation within one facet.

methods **(RQ1.2)** identified in industry practices, as defined in 2.1. The third and fourth facet are closely related to the classification of the teaching approaches **(RQ2.1)** and primary stakeholders **(RQ2.2)** from both industry and education. The last two research types, and contribution type facets, are proposed by Shaw (2003) and Wieringa et al. (2006), respectively.

3.2.4.1. Common SE Trends. We based our work on the initial list of SE Trends reported in Section 2.1. To extend the list, we read the research scope, aim, and stated goals. When abstracts did not provide sufficient information, we also read the introduction and research background sections. The process is similar to the creation of the other facets, mentioned in Section 3.2.4. However, keywords were more difficult to identify because concepts must have clarification and grouping associated with the software industry and education intersections of SE Trends. The final list of SE Trends used in this paper is shown in Table 7.

The rest of the SE Trends from Table 1 (e.g., Computational plenty, Software Engineering for autonomous systems, Biology and computing) are not present in Table 7; we excluded the information either because there was no evidence found during the search or the trends have not yet become emergent in the SE context.

3.2.4.2. Models, processes, and methods in SE. To provide the detailed characteristics of SE Trends, we developed a list of models, processes, and practices that are reported in primary studies.

Table 7
Keywords for common SE Trends.

Category	Properties
Agile Software Development	Agile practices, Lean and Agile, rapid prototyping, Enterprise Software Integration, DevOps, continuous software integration
Software Implementation, Usability and Value	User-centered software, software usability, software non-functional requirements, software value, functionality development
Global Software Engineering	Global software engineering, multinational vs. local environment, cross-site, same-site software development, distributed development
Lean Software Startup	Innovative practices in industry through Tech Startups, Lean Software Startups, BizOps, Tech Startups, Software Startups
System of Systems	Cloud systems, mobile systems, System of Systems, crowd-sourcing, open source software

Applying a similar keywording process, we identified terms for SE Trends' features as shown in Table 8.

3.2.4.3. SEE teaching approaches. To identify the different teaching approaches discussed in each paper, we focused on the methodology described in the abstracts. When abstracts provided insufficient information regarding the teaching strategy used, we also read the introduction and, occasionally, the research

Table 8
Keywords for SE model or process.

Model process method	Keywords
Agile	Agile, development, prototype, industry practices, programming, practices, project management, teams, techniques
Kanban	Kanban
XP	Extreme programming, XP programming
Scrum	Scrum, human-centered, software development
TDD	Test driven development, test case oriented development
Prototype	Prototype-centric, prototype
Lean	Software delivery, software development, learning approach
Other	Other models or processes (Waterfall, Spiral, ad-hoc etc.)

methodology sections. We were able to identify five categories, as shown in Table 9.

3.2.4.4. Stakeholders. Different stakeholders can be identified from both education and industry. We divided the stakeholder categories into two major groups:

- *Industry:* Commonly, contributors from the industry assigned adjunct positions in education. In contrast, researchers who accept research and development roles (R&D) in the industry, utilizing proxies such as research institutes. Furthermore, project managers, customers, and product owners from industry, who are actively participating in capstone and customer-driven courses and adopting Agile/Scrum methodology (Rico and Sayani, 2009; Marques et al., 2018; Paasivaara et al., 2018).
- *Education:* Lecturer, teacher, and professor are interchangeably used terms for identifying an instructor role within SEE courses (Fernanda et al., 2018; Williams et al., 2015). In some cases, when the teaching involves industry training, the instructor represents a professional role from industry (Scharff and Verma, 2010).

After reviewing the abstracts, we can create the facets and their corresponding keywords for classifying the papers. The keywords used to classify the papers into these categories are listed in Table 10.

3.2.4.5. Research type. We based the research type facet on the schema proposed by Wieringa et al. (2006). It contains six categories:

- *Validation research:* Researchers do not implement validation research in practice and focus on the validation of the solution in the lab or simulation scenarios. It is common in SEE validation studies to state hypotheses that use summary statistics to describe the main components of an experimental setup and to include discussions concerning limitations.
- *Evaluation Research:* Researchers implement evaluation research in practice. In the SEE context, it is common to present the solution implementation and argue its consequences in terms of benefits and drawbacks. As stated by Petersen et al. (2008), we can exclude evaluation research if no industrial cooperation or real-world project is part of the study.
- *Solution proposal:* Researchers provide a solution proposal as a new technique or an extension of an existing one. In SEE context, the proposals are limited to arguing the technique benefits, although there is no complete validation within the study.
- *Philosophical proposal:* Researchers describe their point of view regarding the subject while sketching a new way of looking at existing knowledge, without the preciseness of a solution proposal. In SEE, philosophical proposals are in the form of taxonomy or conceptual framework and are usually associated with curriculum change proposals.
- *Opinion paper:* Researchers report their opinion and provide recommendations. In SEE context, opinion papers cover mainly theoretical studies, with some supporting evidence.
- *Experience paper:* Researchers in experience paper in SEE report on personal experiences from a real-life project. The authors do not include research methodology.

We list the keywords used to classify the papers into these categories in Table 11.

3.2.4.6. Contribution type. We adopted the contribution type categories from Shaw (2003), using the following for this mapping study:

- *Model:* Representation of an observed reality by concepts or related concepts after a conceptualization process.
- *Theory:* Construct of cause–effect relationships of determined results.
- *Framework/methods:* Models related to constructing software or managing development processes.
- *Survey:* Empirical investigation through questionnaires, interviews.
- *Guidelines:* List of advice, synthesis of the obtained research results.
- *Lesson learned:* Set of outcomes, directly analyzed from the obtained research results.
- *Advice/implications:* Discursive and generic recommendation, deemed from personal opinions.
- *Tool:* Technology, program, or application used to create, debug, maintain, or support development processes.

The papers' classification is according to the keywords relevant to each category presented in Table 12.

3.2.5. Data extraction and mapping

After defining the classification schema, resulting from the keywording process, we proceed to extract data from the primary studies systematically. We analyze the selected studies in the pool and identify a list of attributes connected to the previously constructed categories. We then store the extracted studies into a systematic map that we use to answer each of the RQs.

To this end, we used the BibDesk tool for categorizing and grouping papers within the different classification schema based on an iterative approach guided by the one proposed by Petersen et al. (2008) and other previous similar literature review studies (Dingsøyr et al., 2012; Garousi et al., 2016; Beecham et al., 2017a). We took good care in checking the following attribute from each paper source:

- Title
- First author
- Year of publication
- Abstract
- Keywords
- Full text (occasionally)
- Publication source

Table 13 shows the data extraction type connected to our RQs. The first column of the table specifies the RQ that is addressed. The second column is connected to the facet or paper attribute. The third column represents the full set of values for each facet

Table 9
Keywords for teaching approaches.

Teaching approach	Keywords
Project based learning	Real world client, user centered design, customer communication, customer participation, localized, open source, free software, Lean Software Startup projects, challenge based learning, capstone projects
Gamified learning	Gamified learning, play money, simulation
Blended learning	Blended learning, online learning, remote courses, MooC
Experiential learning	Team skills, soft skills, experiential learning, autonomous learning
Other	Other learning/teaching approaches that do not fall under the mentioned groups

Table 10
Keywords for stakeholder.

Stakeholder	Keywords
Education	Student, millennial, graduate, undergraduate, learner, undergrad, teacher, professor, instructor
Industry	Client, real-world client, buyer, project manager, product owner, tutor

Table 11
Keywords for research type.

Research type	Keywords
Validation research	Test, result, simulation, emulator, analysis, experiment, prototype
Evaluation research	Evaluation, implementation, result, platform, case study, production
Solution proposal	Solution, proposal
Philosophical paper	Philosophical paper
Opinion paper	Discuss, survey, suggests
Experience paper	Experimental

Table 12
Keywords for contribution type.

Contribution type	Keywords
Model	Model, concepts, process, conceptualization of teaching strategy
Theory	Theory, cause–effect
Frameworks/methods	Framework, architecture, implementation, scheme
Survey	Questionnaire, interview, empirical
Guidelines	Advice, synthesis, best practices
Lessons learned	Outcomes, research results
Advice/implications	Discursive, generic, personal opinion
Tool	Tool, demo, implementation, development, assessment

or paper attribute. Finally, the last column represents the multiplicity of the study contribution to each category. E.g., When the value is "M" (Multiple) for a specific category, it means that the study can contribute to both Agile Software Development and GSE. Whereas when the value is "S" (Single), it means that the study can be part of a specific Research Type, e.g., Validation Research. A similar approach is also adopted from Garousi et al. (2016).

While focusing on each RQ, authors initially reviewed the title, abstracts of the primary studies, and assigned them to different categories. When the categorization was not possible from reading title and abstract, then a full paper text is considered. The process of reading the entire paper eventually required data to be easily identified, tracked, and linked to the RQs from the primary sources. Thus, to explicitly link our study and the categorization of primary sources, the authors started placing color-coding inside the paper pdf files and later developed a summary of the individual primary studies. Fig. 7 shows the color-coding, of one of the primary sources (Rodríguez et al., 2018). Overall, the color-coding helped in tracing of terms and peer-reviewing of the papers, from the authors, while placing them inside the map. In the upcoming section, we utilized the map to answer the RQs.

4. Results

From an initial sample of 1930 papers, we identified 126 primary relevant studies (Appendix) for answering our RQs. Findings answered the primary RQ, as well as its corresponding sub-questions. Furthermore, we checked and reported publication trends and quality in the study. The mapping of individual studies to categories from each facet is presented in Appendix.

4.1. Answering RQ1 - To what extent are *SE Trends* presented in *SEE research*?

To help answer the first RQ, we formulated three ancillary sub-questions. The first sub-question identifies the distribution of SE Trends in the literature, whereas the second sub-question addresses the models, processes, and methods embraced in SEE. Finally, the third sub-question identifies the evolution over time of SE Trends in a SEE setting.

4.1.1. Answering RQ1.1 - The distribution of *SE Trends* in *SEE research*

We present the distribution results[1] of the primary studies in Table 14. It is worth noting that some of the studies fall under more than a single category if multiple common SE Trends are part of the study itself. Thus, the percentage of the appearing results is slightly higher than the overall total number of primary studies (126), with approximately 20% of the papers addressing more than one common SE Trend.

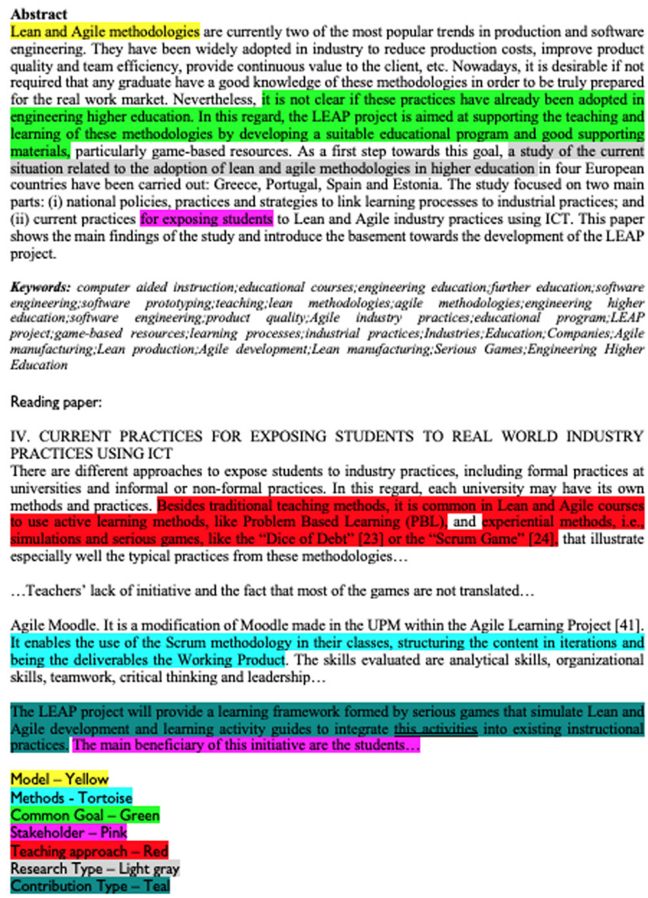
Agile software development represents the most investigated and explored trend within the education context, comprising 79.4% of the overall published papers. Agile Software Development received support, within the course setting, by several different practices (Scrum, Kanaban, XP, TDD) well known in both industry and education (Blasquez and Leblanc, 2018; Ahmad et al., 2014; Murphy et al., 2008; Kollanus and Isomöttönen, 2008; Vu et al., 2009). Combining the practices also seems to be one of the appropriate approaches in various publications (Delgado et al., 2017; Kruchten, 2011). Since the focus of Agile Development is on rapid and adaptive response to change and communication among stakeholders, many of the SE courses involve external industry stakeholders and focus on team soft skills (Rodríguez et al., 2016), as also discussed in 4.2.

Software implementation, usability, and value is another far-reaching SE Trend that has been explored (16.7%) in SEE context. Many of the publications emphasize the software relevance and benefit of its operation, at some particular level, within the course setting (Murphy et al., 2017; Brügge and Gluchow, 2012). The three software dimensions (implementation, usability, and value) within SEE are mostly reflected in course projects requested by external industry actors (Murphy et al., 2017) and sometimes from departments within the University (Liew, 2013).

Lean Software Startup is also fairly significant (9.5%) in the SEE context as part of education for Millennials (Heggen and Cody, 2018), having indicators as a newly emerging strategy. Publications (Devadiga, 2017; Buffardi et al., 2017b), who emphasized

[1] The full map of Common SE Trends is present in Appendix, Table 22.

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.



Abstract
Lean and Agile methodologies are currently two of the most popular trends in production and software engineering. They have been widely adopted in industry to reduce production costs, improve product quality and team efficiency, provide continuous value to the client, etc. Nowadays, it is desirable if not required that any graduate have a good knowledge of these methodologies in order to be truly prepared for the real work market. Nevertheless, it is not clear if these practices have already been adopted in engineering higher education. In this regard, the LEAP project is aimed at supporting the teaching and learning of these methodologies by developing a suitable educational program and good supporting materials, particularly game-based resources. As a first step towards this goal, a study of the current situation related to the adoption of lean and agile methodologies in higher education in four European countries have been carried out: Greece, Portugal, Spain and Estonia. The study focused on two main parts: (i) national policies, practices and strategies to link learning processes to industrial practices; and (ii) current practices for exposing students to Lean and Agile industry practices using ICT. This paper shows the main findings of the study and introduce the basement towards the development of the LEAP project.

Keywords: *computer aided instruction;educational courses;engineering education;further education;software engineering;software prototyping;teaching;lean methodologies;agile methodologies;engineering higher education;software engineering;product quality;Agile industry practices;educational program;LEAP project;game-based resources;learning processes;industrial practices;Industries;Education;Companies;Agile manufacturing;Lean production;Agile development;Lean manufacturing;Serious Games;Engineering Higher Education*

Reading paper:

IV. CURRENT PRACTICES FOR EXPOSING STUDENTS TO REAL WORLD INDUSTRY PRACTICES USING ICT
There are different approaches to expose students to industry practices, including formal practices at universities and informal or non-formal practices. In this regard, each university may have its own methods and practices. Besides traditional teaching methods, it is common in Lean and Agile courses to use active learning methods, like Problem Based Learning (PBL), and experiential methods, i.e., simulations and serious games, like the "Dice of Debt" [23] or the "Scrum Game" [24], that illustrate especially well the typical practices from these methodologies…

…Teachers' lack of initiative and the fact that most of the games are not translated…

Agile Moodle. It is a modification of Moodle made in the UPM within the Agile Learning Project [41]. It enables the use of the Scrum methodology in their classes, structuring the content in iterations and being the deliverables the Working Product. The skills evaluated are analytical skills, organizational skills, teamwork, critical thinking and leadership…

The LEAP project will provide a learning framework formed by serious games that simulate Lean and Agile development and learning activity guides to integrate this activities into existing instructional practices. The main beneficiary of this initiative are the students…

Model – Yellow
Methods - Tortoise
Common Goal – Green
Stakeholder – Pink
Teaching approach – Red
Research Type – Light gray
Contribution Type – Teal

Fig. 7. Data extraction sample from one of the primary papers' summary (Rodríguez et al., 2018). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 13
Data extraction type with respect to RQ.

RQ	Facet/Paper Attribute	Categories	Multiplicity
RQ1.1	Common SE Trend	Categories list in Table 7	M
RQ1.2	Models/Process/Methods	Categories list in Table 8	M
RQ 2.1	Teaching Approaches	Categories list in Table 9	M
RQ 2.2	Stakeholders	Categories list in Table 10	M
RQ3	Research and Contribution Type	Categories list in Tables 11 and 12	S
RQ4	Publication Source	Journal, Conference, Workshop	S

Table 14
Number of papers by education goal in SEE.

Common SE Trends	Number of papers	Percentages
Agile Software Development	100	79.4%
Software Implementation, Usability and Value	21	16.7%
Global Software Engineering	12	9.5%
Lean Software Startups	12	9.5%
System of Systems	2	1.6%

the realistic education setting obtained through this approach. The Tech Startup model experimented with software engineering and entrepreneurship students in Buffardi et al. (2017b), where authors claim a further contribution to new Tech Startup formation. Moreover, there are occasional collaborations that emerge between industry and education, contributing to Startup development (Nguyen-Duc et al., 2016).

Global Software Engineering is another SE Trend encountered in the literature equal to Lean Software Startups. Beecham et al. reported an imminent need to address this research area in late publications (Beecham et al., 2017b,a). Further, the authors suggested how Agile, through Scrum methodology, can adapt to GSE (Bosnić et al., 2015).

System of Systems is one of the least explored trends (1.6%) in the education context. One case mentions, as a pilot study, crowdsourcing (Dow et al., 2013) within the classroom setting and reported three key challenges regarding the student context in relation to how to (1) set expectations, (2) enable deeper interactions, and (3) handle uncertainty. In another case, the development of cloud and mobile solutions is presented with a focus on improving the project management skills via the System of Systems (Neyem et al., 2018).

Compared to the list of trends from Table 1, we notice that some are still left out and present no traces in the current state of SEE (e.g., Computational Plenty, SE for Autonomous Systems and Biology Computing).

Key findings:

1. Almost 80% of the primary studies relate to the most common SE Trend, namely Agile Software Development.
2. Other SE Trends, such as Lean Software Startup, Global Software Engineering, Software Implementation, and Usability and Value, are shown in less than 20% of SEE studies per trend.
3. The SE Trend System of Systems is the least explored, represented by less than 2% of the sources in the SEE research context.

During the investigation process, we identified the major intersection areas between the software industry and SEE, as reported in 2.1. This intersection helped us evaluate the distribution of the SE Trends addressed in the SEE context. To analyze the intersection among the SE Trends more assiduously, we investigated further which practices are presented and which stakeholders are actively participating in the education context. To this end, we have presented the outcomes from the remaining two sub-questions in the upcoming sections.

4.1.2. Answering RQ1.2 - Software industry ***models, processes and methods*** in ***SEE research***

To answer this question, we utilize the model/process and method facets. The most recently adopted model in industry and education is Agile, commonly combined with Scrum methodology. The Agile model is reported in most of the reviewed publications. The context in education is multifaceted, including GSE (Paasivaara et al., 2015; Sievi-Korte et al., 2015), capstone projects (Bastarrica et al., 2017) and Lean Software Startup (Buffardi, 2018).

Some cases (Buffardi, 2018; Rodríguez et al., 2018) combine the Lean approach with the Agile in the education setting. Prototyping is primarily exploited when gamification is involved in the teaching approaches (Pirker et al., 2016). Other approaches include external activities, e.g., hackathons, run for short periods (Nandi and Mandernach, 2016) or tutoring with cognitive systems (Müller et al., 2018). Scrum is one of the most adopted methods, appearing in around 50% of the publications. Papers reported Scrum challenges when teaching or conducting GSE in distributed software development projects (Bosnić et al., 2015). In Startup formation, either in education or industry, Scrum is one of the preferred methods (Bosch et al., 2013b). And, due to the nature of the method in the education setting, it is mainly used in project-based learning (PBL), relying on different teams (Uskov et al., 2016; Wallace et al., 2012). Other similar teaching approaches where scrum is commonly utilized are customer, innovation, and capstone driven courses (Brügge and Gluchow, 2012; Buffardi, 2018; Knudson and Radermacher, 2011).

Methods such as Kanban and XP, although less used, have been successfully adopted in education (Bastarrica et al., 2017; Ahmad et al., 2014). A combination of the Kanban practice with Scrum is reported in Matthies (2018). The mixing of the methods provides further benefits for conducting SE practices by improving implementation quality, efficiency, and final product delivery and usability (Patil and Neve, 2018). Test-Driven Development (TDD) method, to some extent, provides useful benefits in education, involving project development based on storytelling (Christensen, 2009), gamification (Blasquez and Leblanc, 2018), and emphasizing student-centered learning (Buffardi and Edwards, 2012).

Other ad-hoc methods received exploration in education when the prototyping approach is followed (Baldauf et al., 2017).

Key findings:

1. The most common methodological approach in the SEE context is a customized version of Scrum, that fits to specific SE contexts.
2. Various SEE contexts have been reported, i.e., traditional SE courses, capstone projects, distributed projects, and incubators.
3. There are reports about Agile methodologies in other trends, such as Global Software Engineering and Lean Software Startup.

During the investigation, we observed that most recent publications emphasized the use of Agile/Scrum as an important trend for updating SEE curricula. However, a gap still exists in this area since mixing the practices, Agile and Lean or Scrum and Kanban, lack extensive exploration, and the benefits are not yet understood in either the industry and education context. The key findings in this section helped us in answering RQ1.2.

4.1.3. Answering RQ1.3. The evolution of ***SE Trends*** in ***SEE research*** setting over time

The evolution of SE trends as topics in software engineering education is shown in Fig. 8.

In 2008, Agile Software Development is the main SE Trend explored in SEE. After ten years, we traced the developments of Agile and four other SE trends. Studies about teaching Agile methodologies have a stable growth. In 2018, we find 24 primary studies that explore Agile as a SE trend. Software implementation, usability, and value have reported fluctuating interest over the years. Emerging trends displaying growing interest since 2013 and 2016 are GSE and Lean Software Startups, correspondingly. The number of SEE studies about Global Software Engineering is relatively stable over time, with the peak at 2015. Education papers about Lean Software Startup SE Trend have increased interest since 2016. The remaining, and least investigated common SE Trend is System of Systems, which appears only in the last five years.

Key findings:

1. Software Engineering Education is changing to adapt to industrial movements.
2. Agile Software Development is a popular SE Trend and still increasing.
3. GSE and Lean Software Startups are emerging SE Trends in SEE.

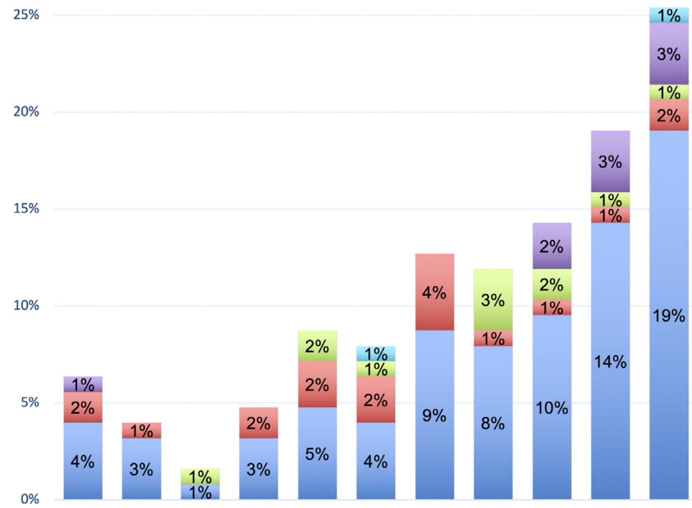
4.2. Answering RQ2 - How does ***SEE research*** present the teaching of various ***SE Trends***?

To help answer our second RQ, we have formulated two ancillary sub-questions. The first sub-question identifies the teaching approaches for SE Trends in SEE context, whereas the second sub-question identifies the stakeholders participating in SEE context.

4.2.1. Answering RQ2.1 - ***Industry-relevant teaching*** approaches presented in ***SEE research***

A summary of distributions of teaching approaches identified from SEE research[2] is presented in Table 15.

2 The full map of Teaching Approaches is present in Appendix, Table 23.



	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Agile Software Development	5	4	1	4	6	5	11	10	12	18	24
Software Implementation, Usability and Value	2	1	0	2	3	3	5	1	1	1	2
Global Software Engineering	0	0	1	0	2	1	0	4	2	1	1
Lean Software Startup	1	0	0	0	0	0	0	0	3	4	4
System of Systems	0	0	0	0	0	1	0	0	0	0	1

Fig. 8. Evolution of the focus from SEE research on SE Trends over time.

Table 15
Number of papers by teaching approach in SEE.

Teaching approach	Number of papers	Percentages
Project based learning	104	82.5%
Gamified learning	6	4.8%
Blended learning	7	5.6%
Experiential learning	1	0.8%
Other	8	6.3%
Total	126	100%

One of the commonly used approaches in SEE research is project-based learning (PBL), which applies different strategies: studio-based learning (Bull et al., 2013; Bull and Whittle, 2014; Lee et al., 2015), customer-driven utilizing open source software (Bruegge et al., 2015; Buffardi, 2017), capstone project courses (Neyem et al., 2014; Paasivaara et al., 2018), local or global scale projects (Paasivaara et al., 2015), and Lean Software Startup oriented projects (Davis and Bolen, 2016; Rico and Sayani, 2009; Buffardi, 2018; Buffardi et al., 2017b; Bosch et al., 2013a).

In the former case, the authors argue that studio-based learning can serve as an approach to energize software engineering education with real-world software engineering practices. The studio approach provides both a lab as well as a means of teaching software engineering. The students can utilize open working environments which provide realistic learning outcomes with eased access to industry setting within the classroom (Kopczyńska et al., 2012). In most cases, we find examples of studios implemented in SEE with the active involvement of industry partners (Root et al., 2008; Kopczyńska et al., 2012; Rosca, 2018; Lee et al., 2015), serving roles such as customer, client, tutor, project manager, and stakeholder. Agile software development based on Scrum or XP is also common trend in studio approaches as reported in Bull et al. (2013), Prior et al. (2014), Lee et al. (2015) and Lattanze (2016).

Whereas, in the latter cases, the authors argue that SE students should be combined with entrepreneurship students on interdisciplinary teams that act on a Tech-Startup driven model. The authors argue that the approach, despite its challenges (intellectual property, legal), may be an emerging candidate for motivating students to deliver more realistic products. It promotes the recommendations of the Agile Manifesto (Beck et al., 2001) of face-to-face communication and external pressure. Experiential-based learning also receives consideration as part of the teaching strategies. Pappas et al. (2018) reflect the benefit of the strategy within the classroom setting. Whereas, Ahmad et al. (2014) utilize the same methodology in a slightly different environment, such as a software factory. In both cases, focus is put on soft skills developed by the students. Moreover, there are cases when a combination of approaches (industry customer, challenge-based, and innovation-driven) is utilized (Llopis and Guerrero, 2018). Blended learning is also present on many occasions, where Massive Open Online Courses (MOOCs) represents one commonly adopted strategy (Xiao and Miller, 2014; Wong, 2016). The use of MOOCs makes its appearance also in a studio-based learning context (Billingsley and Steel, 2014). Another publication describes the use of cognitive systems as part of the teaching methods (Müller et al., 2018).

Key findings:

1. SE trends are mainly addressed by Project-based learning approaches in SEE.
2. Other potential learning approaches, i.e., gamified learning and blended learning, still lack significant exploration.

No matter the teaching approach used, the relevance of skills obtained in education, related to the industry, is a significant

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 16
Number of papers by stakeholder involved.

Stakeholder	Number of papers	Percentage
Education	100	79.4%
Education and industry	26	20.6%
Total	126	100%

indicator for proposed teaching/learning approaches. Many publications discuss this aspect as part of their research motivation. Often, the acquiring of soft skills within the classroom setting is an essential aspect in the Lean Software Startup context. However, we argue that it is not enough to identify the teaching approaches and strategies but also to scope the models, processes, and methods used in connection to them.

4.2.2. Answering RQ2.2 - Stakeholders in SE Trends

To answer RQ2.2, we reviewed most of the publications. In order to observe the level of the stakeholder collaboration from both industry and education, we used the stakeholder facet. Results[3] are shown on Table 16.

In education, the primary actors are education instructors, teachers, coaches, and students identified from capstone courses relying on project-based learning, GSE, and customer-driven courses. In an industry setting, the stakeholders involve mainly developers, project managers, product owners, industry professionals, project leaders, and other external actors, such as clients/customers (Zazworka et al., 2010; Reddaiah et al., 2016). Usually in a course setting, if an industry role is missing, it is either covered by internal university staff acting as clients (Stephenson et al., 2016) or the course instructors. This simulation is only partially effective, due to the lack of real external pressure. We observe that studio-based learning the customer role is frequently covered by the course tutors or academic supervisors, as in Kopczyńska et al. (2012). However, we notice studio cases that report active collaboration with industry stakeholders (Prior et al., 2014; Lee et al., 2015). The industry stakeholder covers the mentor or observer role by participating in weekly studio sessions.

Repeatedly, a recent proposal is made by Heggen and Cody (2018), where the course blends internship and summer jobs within SEE, in order to provide students with appropriate soft and technical skills, while in close contact with many different stakeholders. Moreover, it is essential to understand that real projects have many constraints and involve more actors. Solutions based on prototyping (Kropp and Meier, 2016) may not be enough. Furthermore, software end-product quality and maintenance are essential aspects of SE. Few considered the impact of the practical element of the course settings in delivering useful final products. The proposed models should take care that simulations provide realistic, stakeholder participation in obtaining valid learning enhancement.

We observed, although with lower frequency, that there have been joint efforts of stakeholders from both education and industry that provide more realistic outcomes (Brügge and Gluchow, 2012; Bruegge et al., 2015). Tech Startups, moderately investigated for millennial students, as reported by Buffardi (Buffardi, 2018; Buffardi et al., 2017b; Buffardi, 2018), involve stakeholders mainly from education. Moreover, as previously mentioned, the Lean Software Startup, being an emerging SE Trend in the industry, is introducing new stakeholders (entrepreneurs, innovation centers, and accelerators) as part of SE courses.

Table 17
Number of papers based on contribution type.

Contribution type	Number of papers	Percentage
Model	8	6.3%
Theory	0	0%
Framework/method	23	18.3%
Survey	71	56.3%
Guidelines	7	5.6%
Lessons learned	12	9.5%
Advice implications	0	0%
Tool	5	4%
Total	126	100%

Key finding:

- The actual participation of Industrial stakeholders in SEE is still limited

These findings, alongside the previous ones, help in understanding a complete intersection overview between the software industry and SEE trends. However, to complete the full picture, it is necessary to map based on the findings of the last two facets (research and contribution type), discussed in the following section.

4.3. Answering RQ3 - How do SE Trends contribute to literature?

The types of research contributions and their distribution results[4] regarding SE Trends in education are presented in Table 17.

When considering the classification based on contribution type, surveys are the most commonly encountered contributions, making up 56.3% of publications. Most surveys primarily explore Agile Software Development practices. One example from Bastarrica, Perovich and Samary (Bastarrica et al., 2017) analyzed the outcome of capstone projects for students.

The other mainly encountered contributions related to frameworks/methods encountered in 18.3% of publications. Most frameworks/methods have validation within the classroom setting and experiments (Browning and Sigman, 2016; Chanin et al., 2018). In these cases, authors such as Holmes et al. (2018) propose experimenting with Free Open Source Software (FOSS), while Bruegge et al. (2015) emphasizes the importance of conducting customer-driven courses with real industrial clients.

Model contributions, accounting for 6.3% of the sources, are implemented in real-life case studies with positive outcomes. For example, Rekha and Diniz (Smrithi Rekha and Adinarayanan, 2014; Diniz et al., 2017) proposed FOSS in conjunction with industry projects to enhance students' soft and technical skills. Bollin et al. (2018) proposed the use of a maturity model adopted from industry within a software engineering course. Many similar proposals showed a shift in the models selected, long utilized in the industry within the SEE curricula. Other sources experiment with Lean learning (Chatley and Field, 2017), distributed development in the GSE context (Bosnić et al., 2015), as well as Tech-Startup approaches (Nguyen-Duc et al., 2016; Buffardi et al., 2017b). In industry, context model proposals try to improve current practices in adopting Agile, Lean, and Prototyping approaches, where a common ground of collaboration with education is found.

[3] The full map of Stakeholders is present in Appendix, Table 24.

[4] The full map of Contribution Type is present in Appendix, Table 25.

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

The other papers involve guidelines and lessons learned, derived from either previous experiences or research evaluations, consisting of approximately 9.5% of the sources. Guidelines from stakeholders, coming from industry and education, consist of 5.6% of the sources. Both lessons learned and guidelines, involved stakeholders from both entities (Kruchten, 2011; Krusche and Alperowitz, 2014c). More effort is required in facilitating and standardizing the collaborations, even when best practices and experiences are present.

Approximately 4% of the papers are related to tools, involving gamification (Bartholomew, 2017), intelligent tutors (Dahotre et al., 2011), cloud-mobile based tools (Neyem et al., 2018), and online learning (Williams et al., 2015).

No publications are related to theories or advice implications.

Research type distribution results[5] summary is presented in Table 18.

Classification, based on research type, indicates that many publications are related to validation research (67.4%), followed by evaluation research and solution proposals. Prototyping solutions have been proposed, and students attempted to validate the prototypes within isolated classroom environments, e.g., Duvall (Duvall et al., 2018) validated Scrumage within a class experiment. Others, Paasivaara and Damian, simulated scenarios in GSE by setting reliable, minimum, experimental constraints and exploiting different approaches (Agile and Scrum) for capstone projects (Paasivaara et al., 2015; Damian et al., 2012). However, many case studies evaluate real-life scenarios by conducting them in correlation with industry demands and actual implementations. They mainly represented evaluation research (11.1%) conducted in collaboration with industry. For example, Venson et al. (2016a) emphasized real case study projects that included education-industry collaboration effects for software engineering graduates. This study is useful for justifying why new collaborations driven by industry trends should also be made available to education in a progressive manner. Other publications implemented collaborations through industry instructors (Stephenson et al., 2016) or reported experiences by considering internal university entities as customers (Anslow and Maurer, 2015), within capstone courses. Further, both industry and education settings reported project management teaching experiences (Kruchten, 2011).

Philosophical and opinion papers are not present in the publications.

To observe the relations between the different facets, we mapped the results for one facet against another and presented the outcomes in bubble plots. Fig. 9 displays the distribution of the common SE Trends with respect to research and contribution type. This map allows us to pinpoint how the studies investigated these trends in SE.

We compare common SE Trends to the contribution and research types. Most papers describe Agile Software Development through models or methods combined with validation research. Although the GSE and Lean Software Startups are emerging SE Trends, minimal research addressed them, with even less collaboration between industry and education. This clear gap necessitates consideration in the future. Education and industry stakeholders find common collaboration grounds when evaluation research utilizes surveying.

5 The full map of Research Type is present in Appendix, Table 26.

Table 18
Number of papers based on research types.

Research type	Number of papers	Percentage
Validation research	85	67.4%
Evaluation research	14	11.1%
Solution proposal	8	6.4%
Philosophical paper	0	0%
Opinion paper	0	0%
Experience paper	19	15.1%
Total	126	100%

Table 19
Selected papers sources.

Source	Number of papers	Percentage
Journals	15	12%
Conferences	101	80%
Workshops	10	8%
Total	126	100%

The most common contribution type is surveys, which relies on validation and evaluation research.

Key findings:

1. In SEE, surveys compose the largest chunk of contribution types.
2. Framework/methods are also commonly encountered SE Trends.
3. Proper guidelines, addressing SE Trends in SEE context, receive limited consideration.
4. Proper toolsets for supporting the integration of SE Trends in the SEE context are minimally reported.
5. Validation, experience, and evaluation research are the most frequent research types conducted in this context.

4.4. Answering RQ4 - In which bibliographical sources are studies published?

We answer this question by extracting the conference or the journal in which the papers are published. The Journal Citation Report (JCR) (Anon, 2019b) used the recognition and stability of a journal to evaluate with a systematic and objective system. JCR is an evaluation mechanism based on statistical information from the reference data. We checked the quality of the sources based on the Journal Citation Reports (JCR), indicating the Journal Impact Factor (JIF) and the Scimago Journal Rank (Anon, 2019a), as well as the journal or conference H-index. The quality check is valid until 2018. Due to a large number of conferences, we advise the reader to consult the full list reported in Appendix Table 27. Tables 19 and 20, illustrates where the 126 selected papers are published.

The primary sources are conferences and workshops, about 88%, and the remaining, approximately 12%, are research journals. There is a wide range of sources for the publications, but high-quality SE and Computer Science Education conferences and journals provided the most relevant papers for this study. Publications in journals, such as ACM Transactions on Computing Education (TOCE) and Journal of Systems and Software, have provided access to topics on the involvement of industrial clients, as well as efficiently implemented Agile methods and teamwork in SE courses. One good example is Bruegge et al. (2015). Other publications, IEEE Software, put the focus on GSE as an imminent

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

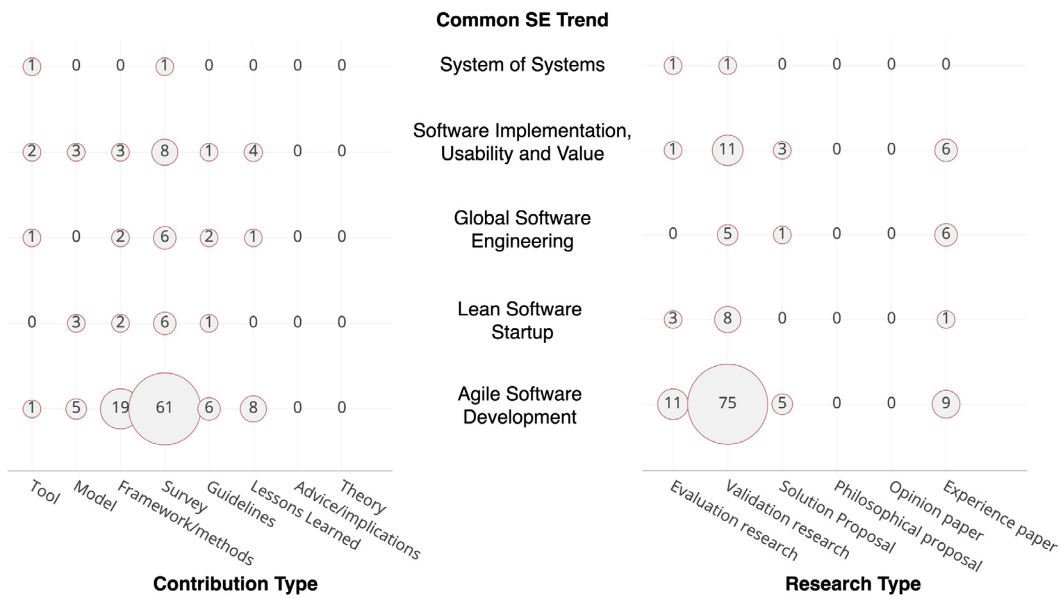


Fig. 9. Number of papers by research, contribution type and common SE Trends facets.

Table 20
Selected papers sources (ordered alphabetically within each category).

	No.	JIF (Anon, 2019b)	H-index (Anon, 2019a)
Journals			
ACM SIGSOFT Software Engineering Notes	1	–	–
ACM Transactions on Computing Education	3	1.356	24
Computer Applications in Engineering Education	1	1.435	24
IEEE software	1	2.945	99
IEEE Transactions on Education	2	2.214	61
Journal of Computing Sciences in Colleges	4	–	–
Journal of Systems and Software	1	2.559	94
Mobile Information Systems	1	1.635	25
World Transactions on Engineering and Technology Education	1	–	10
Conferences			
Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)	11	–	22
Annual ACM SIGITE Conference on Information Technology Education	4	–	5
ACM Technical Symposium on Computer Science Education	7	–	22
Frontiers in Education Conference (FIE)	7	–	32
Global Engineering Education Conference (EDUCON)	9	–	14
International Conference on Software Engineering (ICSE)	17	–	118
International Conference on Software Engineering Education and Training (CSEE&T)	18	–	6
Other conferences with < 3 papers in our map (cf. Table 27 in Appendix for full paper list)	28	–	–
Workshops			
International Workshop on Games and Software Engineering	1	–	–
International Workshop on Software Engineering Curricula for Millennials	4	–	–
European Conference on Software Architecture Workshops	1	–	–
First International Workshop on Software Engineering Education Based on Real-World Experiences	2	–	–
ICSE Workshop on Cooperative and Human Aspects of Software Engineering	1	–	–
Second International Workshop on Collaborative Teaching of Globally Distributed Software Development	1	–	–
Total	126	–	–

need to prepare future SE students (Beecham et al., 2017a) and MOOCs as a tool that can transform SE learning on a global scale (De Freitas et al., 2015).

Lean Software Startups in SE courses (Buffardi, 2018; Chatley and Field, 2017) have been identified as an emerging trend in recent high-level workshop and conference publications, such as ACM/IEEE International Workshop on Software Engineering Education for Millennials in 2018 and the International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET) in 2017.

Key finding:

- Primary studies are published predominantly within high-quality journals and acknowledged conferences.

5. Discussion

In this section, we further discuss the key findings from the sections answering the RQs. We provide further insight into the relevance of the results in comparison to the current state of knowledge.

5.1. Common SE Trends

We analyzed the current situation of common collaboration, practices, models, methods used in industry, and education. Based on the results of this systematic mapping study, we first find that common SE Trends consistently vary in the last five years, with publications from this time period comprising more than 50% of the total papers. Also, software industry sectors are rapidly evolving, and education is acknowledging the need to join forces. Of all the common SE Trends, Agile Software Development appears in 79.4% of the sources (Table 14), followed by Software Implementation, Usability, and Value practices (16.7%). We argue two main reasons for Agile Development having an increased interest in SEE: (1) it has maturity in the industry, and (2) it is an easily comprehended practice in the education context. Although SEE stakeholders are recognizing the value of the software being developed, it remains in a formative phase in this area. Furthermore, there is a clear gap in the adoption of Lean Software Startups and Global Software Engineering trends. As described in the results, teaching Lean Startups is an emerging trend over the last five years. How SEE will respond to these SE Trends may be crucial for future student cohorts. It is not surprising that the main collaboration scenarios between industry and education stakeholders involves Lean Software Startup formation and the GSE context. Moreover, we argue that positive benefits will result from utilizing both trends for improving students' soft and technical skills.

System of Systems is the least explored SE Trend. Education has to invest immediate attention in this area in order to contribute to the continued preparation of future millennial students. A primary reason for this trend's minimal presence in SEE could be a lack of resources that enable students to practice in these systems. Another reason could be the inability of SEE to simulate collaboration with large system providers and developers that provide students practice at the appropriate level.

The list of SE Trends in the industry, Table 1 from Section 2.1, does include and involve some other future areas, such as Computational Plenty, SE for Autonomous Systems, and the combination of Biology and Computing. Unfortunately, it was not possible to find any traces of these trends believed to emerge in the SEE after 2020 or 2025. Thus, we conclude that SEE remains in a stagnation phase while waiting for the industry to make further advances and before embarking on new SE Trends.

We argue that compared to neighboring domains, such as Computer Science (CS) and Information Systems (IS), there might exist commonalities with our present findings. However, to avoid a sterile discussion, we recommend that CS and IS domains look into a similar approach to identify trends in industry and education.

5.2. Research and contribution type

Evaluations of real scenarios are at a meagre level (11.1%), while even fewer publications report on tools (4%). This demonstrates that more effort is required for providing joint solutions and development of collaborative tools. Validations, by way of survey or experiments, are the most encountered research type (67.4%) in the education context. This percentage is greater, by far, than the other research types, demonstrating that most of the results are experimental and not field-tested through real case studies. Very few solution proposals (6.4%) provide scenarios and directions for grounding education and industry collaboration. Moreover, solutions are presented separately from the industry and education contexts. Less than 1% of the solution proposals related to collaboration efforts between industry and education (Brügge and Gluchow, 2012). This critical situation must receive attention in the future, as stated from Garousi et al. (2016).

5.3. Papers evolution over time and publication sources

A certain number of published reports have unstable increase and decrease rates until 2015, after which, we notice a constant publication increase. The years with fewer publications are 2008 and 2010, respectively, with most of the reports (about 60%) published between 2015 and 2018. This increase demonstrates emerging trends have increased interest in the SEE context.

When looking at the evolution of the research type, we observe that the portion of papers concerning validation research have a constant increase over time. However, solution proposals and evaluations, via real-life projects, fluctuate over the years. The growth in validation research, as well as a decreasing number in solution proposal and evaluation research, provides some evidence which suggests that software engineering in the education and industry intersection is moving from fundamental theory to practical validation. This growth could be a sign of burgeoning maturity or a need to demonstrate the validity of the results. Moreover, there exists a need to evaluate the actual implementations with case studies comprised of joint efforts, as reported in Garousi et al. (2016). We admit that other extraneous factors might influence the validation research growth. One example can be the fulfillment of publication venues' expectations for papers, which include validation data. Unconsciously lessening solution proposal research, involving novel course proposals within SEE.

We observed that a good portion of the reports are available in highly ranked journals or conference proceedings. This measure of the reports reinforces the quality of our research and allows conformation of the reliability of this study.

5.4. Difficulties and potential solutions in the adoption of SE Trends within SEE

Through analysis of our research findings, we have identified and grouped various difficulties, proposed adjustments to educational practice, and suggested possible reasons for gaps in adopting SE Trends within SEE. These are displayed in Table 21, which follows. Although we are able to provide a rudimentary list of the problems we encountered based on our findings, creating a complete map of challenges will require meticulous continuing investigation from the research community.

As discussed earlier, the Agile Software Development Trend has made its way in SEE, but not without pending challenges. One of the significant difficulties that Agile Software Development faces is overall course design. Managing both student teams and industry stakeholders is a logistical challenge for those seeking to develop course settings which correspond to and integrate industry norms. When cooperation with industry stakeholders is lacking, we notice little to no external pressure for students to deliver realistic products. Some courses also lack the rigor to appropriately follow the Agile practices as recommended by Schwaber and Sutherland in the Scrum Guide (Schwaber and Sutherland, 2011).

Table 21
Difficulties and solutions in the adoption of SE Trends within SEE.

SE Trend	Difficulties	Solutions
Agile Software Development	– Designing manageable courses – Introducing adequate external pressure – Recreating Agile industry practices – Prepare students with real-challenges	– Integrated course modules (e.g., SE classes and summer jobs) – Longer course duration (e.g., SE courses spanning over two semesters) – Active involvement of industry stakeholders (mentors, clients, and customers)
Software Implementation, Usability and Value	– Emulating industry software development environment in SEE	– Integrating practical knowledge with theory through open lab spaces (cf. studio-based learning), – Active involvement of industry stakeholders (mentors, clients, and customers)
Lean Software Startup	– Fostering innovation mindset within SEE – Adopting innovative technologies and tools	– Include external activities into courses (e.g., Hackathon, Bootcamp, and Software Workshops) – Active involvement of industry stakeholders (mentors, clients, and customers)
Global Software Engineering	– Adopting collaborative development – Tackling cultural differences, tool choice, technical difficulties, and time zone differences	– Frameworks of collaboration among Universities – Active involvement of industry stakeholders (mentors, clients, and customers)
System of Systems	– Coordination and technical set up of the course	– Partnership with enterprises from the software industry

One key finding is that course material and structures tended to skew away from real-world conditions. To address this issue, we recommend the following adjustments. First, course designers should consider integrating SE courses with summer jobs in the industry. This approach would help students get a taste of the Agile practices while tackling challenges in a professional setting. Secondly, SE courses should span two semesters and incorporate the active participation of industry stakeholders. The roles taken by these stakeholder participants can encompass anything from mentors to clients, as already noted by many sources.

Although the Trend of Software Implementation, Usability and Value is covered in many publications, this focus is still underrepresented in literature specially dealing with SEE. A critical reason for this gap could be the difficulty of emulating industry software development environments within SE courses. We observe encouraging results from studio-based learning, customer-driven, and free open-source courses. This last model, however, might not adequately incorporate industry stakeholder involvement. As in the case of the Agile Software Development Trend, we argue that the active participation of industry stakeholders should become commonplace in SE courses.

Lean Software Startup Trend is underrepresented in the SEE context. Difficulties faced while applying it in this context are related to (1) SE students' mindsets when developing innovative ideas, and (2) delays in adopting innovative technologies, tools, and SE practices (we often observe a combination of existing Lean and Agile practices). We argue that the inclusion of external activities such as hackathons, bootcamps, and innovation workshops, which are commonplace in the industry, would help to overcome present difficulties when designing courses addressing this SE Trend. Again, it is noteworthy that active involvement of external stakeholders will drive realistic project ideas within SE courses. However, the difficulty of securing stakeholder involvement does not, in itself, justify the gap in representing this Trend within SEE literature. We argue that the main cause for an existing gap is the lack of a conceptual model that would facilitate Lean Software Startup research to become part of SEE courses. Also, experimenting with new curricular materials is challenging for most educators unless clear benefits can be immediately appreciated.

The GSE Trend is equally represented as Lean Software Startup Trend within SEE literature. We observed difficulties in adopting collaborative software development approaches similar to those found in industry settings. Other issues related to cultural differences, tool choice, technical challenges, and time zone differences generated further issues in implementing GSE-based courses. A framework of curricular collaboration among universities would help mitigate some existing challenges without overlooking the active participation of industry stakeholders. We argue that the main reason that GSE Trend is struggling to emerge in SE courses is due to a dearth of functional collaborative frameworks between academic institutions.

Lastly, System of Systems is the least represented of all SE Trends within SEE. The reasons might vary vastly, and we would prefer sticking to the evidence at hand. The most significant difficulty reported in the literature relates to the coordination and technical setup of systems within SE courses. SE course educators make reasonable efforts to address this SE Trend. Partnership with enterprises from the software industry might help in overcoming the coordination and technical setup challenges. Unfortunately, industry/education collaborations addressing this Trend are still uncommon, as signified by their low representation in SEE literature.

5.5. How can practitioners, researchers and educators use our findings?

From a practitioner's standpoint, the results presented in this study provide an overview of the common interests shared by the software engineering industry and education. The SE area shows a certain level of maturity when it comes to presenting models, methods, and frameworks which have been successfully applied in real case studies. However, increased joint efforts are required to address emerging trends, such as the new focus on Software Implementation, Usability and Value, Lean Software Startups, and GSE Trends. To help practitioners to help and benefit from SEE, students and educators should:

1. Evaluate the potentials of emerging and future SE Trends throughout experimental courses or training sessions within the SEE context
2. Exploit the low-cost software development opportunities in educational contexts where proper focus is put on good implementation practices, product quality, and value
3. Motivate student employment and decrease their skill gap through participation in SE courses
4. Develop new teaching methods based on new SE Trends requirements

From a researchers' point of view, multiple lines of investigation emerge as a result of the gaps found in the mapping. Most of the investigations seem to be based on surveys, with little of the gathered empirical evidence utilized in the studies.

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Based on Tables 17 and 18, there is a need for research on framework/method proposals and tools (only 4% of papers report new tools), as well as more experience and solution proposal papers, which can be further evaluated via real case studies. Stakeholder involvement, from both education and the industry (20.6%), is also a crucial aspect that requires immediate addressing from both contexts. Thus, in the near future, researchers should focus on the following:

1. *Validate emerging trends such as Lean Software Startup and GSE.* We observe from Fig. 9 that significant research type contributions are missing in these areas such as solution proposals, evaluation research, and experience reports. Due to the infancy of both research areas, we argue that the research community can benefit from solution proposals. Focus can be put on proposing and evaluating new teaching approaches that foster SE students' innovation mindsets and potential interests in forming startups, all while tackling global SE challenges in the education context. Conducting further evaluation research would require more industry-education collaboration. From Table 16, we observe the need for an increased presence of industry stakeholders in the education context. Researchers should also be mindful of the industry-relevant teaching approaches presented in Section 4.2.1. We argue that external industry stakeholders from Lean Software Startups and GSE could be quite viable collaborators. Experience reports would also be beneficial to identifying challenges that education is facing in introducing these two SE Trends in an academic context. We argue that the infancy in exploring these trends is partially due to the slowness of academic instruction to embrace new SE Trends. Challenges arise from competencies that students should have mastered, like soft skills such as distributed communication and teamwork, team-building, and multi- and inter-disciplinary collaboration. Furthermore, adherence to Agile and Lean practices in unpredictable startup and distributed contexts will pose real-world challenges that students need to learn to overcome. Even though instructors have emulated industry roles, there is still a need for actual industry stakeholders to put the SE course in more realistic perspective. Adequate collaborative tool choice in SEE for teaching these Trends is still an open debate. Repeatedly, we observe a lack of new tools being contributed. Moreover, the need for new models, frameworks, and methods is more urgent in GSE, since we find some proposals already emerging in the Lean Software Startup context.
   From the already-identified pool of studies, we observe positive results from industry/education collaborations. Our recommendation for overcoming present and future challenges in this area would be to facilitate collaborations in a more structured manner in order to make them more sustainable and easier to plan.
2. *Propose tools that facilitate SE Trend adoption in the SEE context.* While observing the Tool column in Fig. 9, we notice that there is only a meagre number of tools for each SE Trend. The reason could be that specific tools require extensive evaluation. Challenges arise in evaluating how tools can improve student learning, help them accomplish their tasks, promote training effectiveness in academic and industry contexts, and gauge the students' work experience and competences. Researchers should also design new tools bearing in mind that their usability should go beyond the classroom setting and that tools can be maintained and improved within the SEE context. The latter might require dedicated research teams.
3. *Provide guidelines for future research directions.* Only a small subsection (around 5.6%) of papers provide guidelines for eventual research. Most of these papers deal with the Agile Software Development Trend, likely due to the infant maturity of present SEE research in all other SE Trends. We can only hope that with the increase of experience research, we will observe the delineation of further guidelines for each particular SE trend in our list. To move towards this outcome, researchers should continuously actualize themselves on the current areas in need of research for each specific SE trend we have identified. The resulting focused, systematic reviews will help in providing insightful research recommendations. Challenges shall remain for the newly emerging SE Trends in the future. We admit that the scientific process is in any case iterative and requires close attention from the research community.
4. *Propose solutions for adopting mixed SE models and methods.* We observe that mixed approaches are becoming more popular (Rodríguez et al., 2018; Heikkilä et al., 2016; Buffardi, 2018; Matthies, 2018). The strategy of introducing mixed SE approaches (e.g. Agile/Lean, Scrum/Kanban) varies from capstone courses to software studios or factories. Combining Lean and Agile or Scrum and Kanban has resulted in reported positive outcomes with increased student satisfaction. However, the pool of publications regarding this practice is still small. One reason, we argue, is the comfort of many researchers in sticking with standard SE practices. We admit that experimenting with courses requires further collaboration with educators and industry stakeholders. Challenges remain in using combined approaches, such as working with cross-disciplinary teams, developing real industry projects, and collaborating with real industry customers. However, the research community has so far obtained positive student feedback while relying on experimental SE practices. It could be wise for researchers to start proposing frameworks/methods to be adopted by educators utilizing experimental SE practices.

Educators in the future should:

1. *Explore emerging SE Trends, such as Lean Software Startups, GSE, and potential industry trends for the coming decade.* The primary concern raised by SEE research is the urgency of educators addressing these two trends. Damian et al. (2012) provide a comprehensive list of challenges in developing GSE courses (e.g., cultural differences, technical challenges, time zone differences, semester length, course, and curriculum differences). Making a similar point, Buffardi (Buffardi et al., 2017b) emphasizes that the cross-disciplinary team setting, intellectual property rights, and lack of funding in Lean Software Startups courses are common challenges for students in developing useful and realistic products.
   We encourage educators to put challenges specific to the classroom setting in perspective with software industry challenges. We want to stress that the meagreness of industry/education collaborations reported in Table 16 is itself an open issue to be addressed for these two SE Trends.
2. *Provide courses that have more realistic SE settings.* Most of the studies we consulted report on the PBL teaching approach, which enhances the hands-on course experience. A reasonable effort is also made to provide courses with a longer duration (Heggen and Cody, 2018) and a more realistic setting. Educators have overcome the dilemma of defining how much industry stakeholders should participate in their courses. However, we noticed that assigning roles to external industry stakeholders in the course setting

is still challenging. There is no collective agreement in assigning roles, and most of the choices are made based on immediate needs. We previously discussed the randomness of decision-making due to the lack of framework/methods and tool contributions for most of the SE Trends. We argue that once validated, the proposals for frameworks/methods and tools presented from the research community, will not be rejected by educators adopting them for use in a classroom setting.
3. *Foster the collaboration of industry-education joint activities within the SE courses.* Occasionally, we observe the introduction of workshops (Paiva and Carvalho, 2018), hackathons (Nandi and Mandernach, 2016). The most utilized approach remains the PBL, which is largely represented here by studio-based learning (Kopczyńska et al., 2012). We encourage educators to explore similar activities that are often common in industry (e.g. bootcamps, workshops). The adoption of activities to which the software industry is already familiar might encourage further participation in the education context.

Practitioners in the future should:

1. *Consider adopting new SE practices based on present and future SE Trends.* We argue that practitioners can benefit from the large amount of validation research already present in an education setting in Fig. 9. It might be useful for the industry to observe the effectiveness of adopting new SE practices (e.g., the combination of Agile and Lean) and make the knowledge transferable into their software industry context.
2. *Collaborate closely with the educators to develop hybrid SEE curricula where industry participation becomes a requirement in most SE courses.* Although we demonstrate that the industry/education collaboration is still evolving, the chances are that industry stakeholders can trigger a standard in SEE curricula by (a) making industry resources more available (c) dedicating their time and effort to students (c) fostering startup formation and innovation and (d) providing competence and knowledge. We are aware that financial instruments are required to facilitate the collaboration process, but there is no lack of encouraging sample models, such as studio-based learning.
3. *Disclose SE practices early on so that the SE Trends can be naturally adapted to SEE.* A structured framework that would ease the collaboration between industry and education could also promote the disclosure of relevant industry knowledge and Trends otherwise considered secret. This proposal has not yet been implemented in SEE research, leaving open the question of how exactly practitioners can contribute by disclosing early on their practices.
4. *Provide their expectations of future SE employees' skills for every SE Trend to foster industry-aligned learning outcomes in SEE.* One of the essential activities in SEE is learning outcomes for students. However, aligning these with industry requirements is a separate challenge of its own. We propose that practitioners should make a reasonable effort during their collaboration process with educators to articulate the competences they expect from future student cohorts.

5.6. Limitations of the study and threats to validity

We discuss in this section potential threats to the validity of our study, alongside the steps that we have taken to mitigate their impact. We identified the following four categories, each representing a threat, as recommended in Wohlin et al. (2012) and observed from other previous similar literature review studies (Dingsøyr et al., 2012; Garousi et al., 2016; Beecham et al., 2017a):

- *Internal validity*: Internal validity characterizes the extent of our conclusions' causality derived from the extracted data. We describe the systematic approach utilized for article selection in Sections 3.2.2 and 3.2.3. To ensure study replicability, we defined and reported (1) search terms and corresponding explanations; (2) search engines used; (3) search time period; (4) search protocol (meta-data used for the search); and (5) inclusion/exclusion criteria. We admit that potential issues and limitations can arise during the selection process, such as (1) the limitations of search terms, which might lead us to not capturing relevant primary studies; (2) the lack of relevant further investigations available in other digital resources (Elberzhager et al., 2012); and (3) the authors' bias in applying exclusion/inclusion criteria, which could again potentially lead to erroneously discarding relevant primary studies.
  To mitigate search limitations, each author proposed different terms pointing to a similar concept, following PICO criteria guidelines (Kitchenham and Charters, 2007). We combined automated searching with manual searching in comprehensive academic databases, forums, and venues closely related to our field of study, for example active venue tracks such as ICSE-SEET and CSEE&T-Research. According to Zhang et al. (2011), IEEE Xplore and the ACM Digital Library are the main search portals in software engineering. Both the search terms and search engines underwent an iterative refining process spanning the search period of about three months. Therefore, we believe that an adequate and inclusive baseline for determining search limitations was identified for this study. Nevertheless, relevant studies still might have been omitted from our consideration for various reasons: different terminology used by authors coming from different pedagogical perspectives, search string effectiveness, venues considered, and manual search limitations.
  We applied inclusion/exclusion criteria to each primary study subject in our investigation. We admit that the filtering process is tightly linked to the authors' judgment and experience. Therefore, the process suffers from author bias. To minimize such bias, we conducted a peer assessment of each primary study, with disagreement resolution from a third, more experienced author.
  Internal validity also warrants concerns regarding data analysis and data extraction during the systematic mapping of abstracts that can be mitigated by a complete literature review. Whenever abstracts cover all relevant information, such as research context, method, and conclusions, the study classification is simplified. However, during the search and filtering phases, the authors performed a more detailed examination of potential papers that had unclear classifications due to their misleading abstracts. While considering more parts of the studies, the authors gradually increased the validity of the obtained results. To ensure that data extraction was adequately achieved, the authors created a classification scheme, utilizing widely accepted guidelines (Petersen et al., 2008; Kitchenham and Charters, 2007).
  Although we made a thorough effort to keep the internal validity of our study strong, we know that some of our results and conclusions might still be skewed based on the final list of primary studies. Risks, as discussed earlier, of missing out relevant studies remain.

- *Construct validity*: Threats related to construct validity in our study are concerned with the suitability of the RQs and the categorization scheme used for data extraction. The authors were cautious about preserving traceability between research goals and questions, which in turn are answered based on the categorization scheme. The categorization scheme was built following guidelines for systematic literature mapping and review studies (Petersen et al., 2008; Kitchenham and Charters, 2007). It underwent an iterative refinement process, as presented in Section 3.2.4. We were careful in matching facets of our scheme to the research sub-questions.
- *Conclusion validity*: Conclusion validity is related to articulating sensible conclusions based on rigorous and replicable treatment. Authors can miss relevant studies or perform incorrect data extraction, which are threats to conclusion validity. Missing or improperly categorizing the studies can lead to the distortion of statistical analysis during the selection phase. The authors were careful in utilizing inclusion/exclusion criteria that permit covering the most extensive number of papers possible as part of the study domain. However, during the selection and classification process, there is still a potential author bias that was mitigated here by carefully describing each component of the research process (Elberzhager et al., 2012), to ensure the correct outcome of the results. Furthermore, all primary source choices were reviewed by at least two authors to mitigate the bias in constructing the categorization scheme and extracting data. Disagreements were managed in collaboration with at least three authors' involvement, primarily relying on consensus. We can argue that in the case that other authors follow the systematic approach and our described procedure, their results and findings will have few significant deviations from ours. We can argue that in case other authors follow the systematic approach, and our described procedure will have little deviations from our results and findings, worth mentioning.
- *External validity*: External validity issues in a mapping study relate to the generality of the results (Easterbrook et al., 2008). Based on the inclusion/exclusion criteria in Section 3.2.3, authors only selected peer-reviewed studies in the English language. Moreover, the authors focused on SE Trends and their applicability in SEE research, as well as on the software industry, thus drawing conclusions only for this area of investigation. Also, when identifying SE Trends, we focus on SE practices and processes. A broader scope of the investigation and a higher granularity level of SE practices increase the risk of the collected data being unmanageable. Maintaining a proper focus on our research reduces the generalizing of outcomes. We rely on the assumption that all proposed industry Trends are to be part of the SEE despite their popularity. Other emerging trends that are not mentioned from the existing research community and that are still part of the gray literature or white papers are not considered. Our assumption and human lack of capability to identify emerging trends pose some risk to our study. Hence, minor threats to external validity in this study context remain.

6. Conclusions and future work

In this paper, we reported a systematic mapping study of empirical Software Engineering Education papers written about Software Engineering Trends. The RQs helped us identify the common SE Trends in industry and how they evolved in academic settings. The authors conducted a thorough analysis by reading a variety of papers and classifying them based on common SE Trends, research type, contribution type, and stakeholder facets. We find the most popular SE Trend in education, namely Agile Software Development, and followed by Software Implementation, Usability, and Value. Other SE Trends, such as Lean Software Startup and Global Software Engineering, present in a smaller range of SEE studies but continue to show growth over the last five years. Thus, reflecting a gap that necessitates attention from researchers and educators. The SE Trend System of Systems is little explored in the SEE context. We also find that Scrum practices are the most common methodologies. Efforts are made to combine Agile with Lean methods, and GSE.

The integration of SE Trends is primarily addressed by Project-based learning approaches in SEE. Other potential learning approaches, i.e., gamified learning and blended learning, receive minimal exploration. We observe that the actual participation of Industrial stakeholders in SEE remains limited; thus, impacting the tendency for the adoption of new SE Trends, which can lead SEE into a stagnation period. The research types primarily focused on validation and evaluation, which comprises the majority of the studies. There remains a requirement for solution proposals and experience papers. Opinion and philosophical papers may lack relevance in the area of investigation, which limited their presence in the publications. In terms of contribution, there is a need for tools and guidelines, since surveying the area and framework/method proposals receive more extensive coverage.

Based on the study findings, we conclude that there is an imminent need for addressing trends such as Lean Software Startups and GSE, due to their growing interest in both industry and education contexts. For both trends, there are numerous potential contributions possible in various research areas. Although this trend is in an early developmental stage, the same potential exists for System of Systems. All the other trends, not yet present in the SEE context, may soon require active consideration too. In the future, we intend to evaluate further how GSE, Lean Software Startups, and SE practices can become parts of SEE.

CRediT authorship contribution statement

Orges Cico: Conceptualization, Methodology, Writing — original draft, Data curation, Investigation, Formal analysis. **Letizia Jaccheri:** Project administration, Funding acquisition, Data curation, Writing - review & editing, Visualization, Supervision . **Anh Nguyen-Duc:** Conceptualization, Visualization, Writing - review & editing. **He Zhang:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is part of the project International Partnerships for Excellent Education and Research in Information Technology (IPIT) project number 274816, funded by the Research Council of Norway (NFR), INTPART from 1.1.2018 to 31.12.2020.

Appendix. Mapping of individual studies to categories

Tables 22–28 present the mappings of the articles to the categorizations presented in Section 4.

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 22
Studies per SE Trend.

Common SET	Studies
Agile Software Development	Ahmad et al. (2014), Ahmad et al. (2014), Bai et al. (2018a), Baldauf et al. (2017), Bartholomew (2017), Blasquez and Leblanc (2018, 2017), Bosnić et al. (2015), Browning and Sigman (2016), Bruegge et al. (2015), Buffardi (2018, 2017), Buffardi and Edwards (2012), Buffardi et al. (2017a,b), Campbell and Tafliovich (2015), Chatley and Field (2017), Choudhari and Suman (2015), Christensen (2009), Corral and Fronza (2018), Damian et al. (2012), Davis and Bolen (2016), Delgado et al. (2017), Devadiga (2017), Duvall et al. (2018), Fagerholm et al. (2017), Felker et al. (2012), Fernanda et al. (2018), Fitsilis and Lekatos (2017), Goto et al. (2014), Heikkilä et al. (2016), Heinonen et al. (2013), Hof et al. (2017), Igaki et al. (2014), Iyengar (2009), Kizaki et al. (2014), Knudson and Radermacher (2011), Kollanus and Isomöttönen (2008), Kropp and Meier (2014), Kruchten (2011), Krusche et al. (2018), Kudikyala and Dulhare (2015), Kuhrmann et al. (2013), de Lange et al. (2016), Liew (2013), Llopis and Guerrero (2018), Lynch et al. (2011), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Marques et al. (2018), Missiroli et al. (2017), Missiroli et al. (2016), Molléri et al. (2018), Murphy et al. (2008, 2017), Nersesian and Spryszynski (2018), Neyem et al. (2014), Olson and Gibbons (2018), Paasivaara et al. (2015, 2014, 2018), Paiva and Carvalho (2018), Palacin-Silva et al. (2017), Pérez-Castillo et al. (2018), Potineni et al. (2013), Rico and Sayani (2009), Rodriguez et al. (2015), Rodríguez et al. (2016), Rodríguez et al. (2018), Santos et al. (2018), Scharff and Verma (2010), Scott et al. (2016), Sievi-Korte et al. (2015), Smith et al. (2011), Soundararajan et al. (2012), de Souza et al. (2015), Stapel et al. (2008), Steghöfer (2018), Steghöfer et al. (2016), Uskov et al. (2016), Villavicencio et al. (2017), Vu et al. (2009), Wallace et al. (2012), Wong (2016), Matthies (2018), Meier et al. (2016), Pirker et al. (2016), Lee et al. (2015), Prior et al. (2014), Bull et al. (2013), Root et al. (2008), Lattanze (2016), Rosca (2018), Chenoweth (2008), Dagnino (2014), Portela et al. (2017), Eddy et al. (2017), Venson et al. (2016b), Bai et al. (2018b), Pérez-Castillo et al. (2018) and Krusche and Alperowitz (2014c)
Software implementation, usability and value	Brügge and Gluchow (2012), Campbell and Tafliovich (2015), Caspersen and Kolling (2009), Dahotre et al. (2011), Felker et al. (2012), Goto et al. (2014), Heggen and Cody (2018), Liew (2013), Murphy et al. (2017), Nersesian and Spryszynski (2018), Pirker et al. (2016), Xiao and Miller (2014), Bull and Whittle (2014), Kopczyńska et al. (2012), Loksa et al. (2013), Billingsley and Steel (2014), Honig (2008), Huang and Port (2011), Rong et al. (2014), Rusu and Swenson (2008) and Penzenstadler et al. (2013)
Global Software Engineering	Bosnić et al. (2015), Buffardi (2017), Damian et al. (2012), Fu et al. (2018), Meier et al. (2016), Paasivaara et al. (2015), Williams et al. (2015), Wong (2016), Paasivaara et al. (2013), Sievi-Korte et al. (2015), Bosnic et al. (2010) and Almeida et al. (2012)
Lean Software Startups	Buffardi (2018), Buffardi et al. (2017a,b), Chanin et al. (2018), Chatley and Field (2017), Davis and Bolen (2016), Fagerholm et al. (2017), Llopis and Guerrero (2018), Nandi and Mandernach (2016), Nguyen-Duc et al. (2016), Rodríguez et al. (2018) and Chenoweth (2008)
Systems of systems	Dow et al. (2013) and Neyem et al. (2018)

Table 23
Studies per Teaching Approaches.

Teaching/Learning approach	Studies
Project based learning	Vu et al. (2009), Rico and Sayani (2009), Knudson and Radermacher (2011), Lynch et al. (2011), Potineni et al. (2013), Ahmad et al. (2014), Kizaki et al. (2014), Goto et al. (2014), Sievi-Korte et al. (2015), de Souza et al. (2015), Kudikyala and Dulhare (2015), Uskov et al. (2016), Nguyen-Duc et al. (2016), Davis and Bolen (2016), Villavicencio et al. (2017), Missiroli et al. (2017), Marques et al. (2018), Fagerholm et al. (2017), Delgado et al. (2017), Palacin-Silva et al. (2017), Llopis and Guerrero (2018), Rodríguez et al. (2018), Fernanda et al. (2018), Almeida et al. (2012), Bai et al. (2018a,b), Blasquez and Leblanc (2017, 2018), Bosnic et al. (2010), Bosnić et al. (2015), Browning and Sigman (2016), Bruegge et al. (2015), Brügge and Gluchow (2012), Buffardi and Edwards (2012), Buffardi et al. (2017a), Buffardi (2018, 2017), Buffardi et al. (2017b), Bull et al. (2013), Bull and Whittle (2014), Campbell and Tafliovich (2015), Chanin et al. (2018), Chatley and Field (2017), Chenoweth (2008), Choudhari and Suman (2015), Corral and Fronza (2018), Dagnino (2014), Damian et al. (2012), de Lange et al. (2016), Devadiga (2017), Duvall et al. (2018), Eddy et al. (2017), Felker et al. (2012), Fitsilis and Lekatos (2017), Fu et al. (2018), Heggen and Cody (2018), Heinonen et al. (2013), Honig (2008), Huang and Port (2011), Igaki et al. (2014), Iyengar (2009), Kollanus and Isomöttönen (2008), Kopczyńska et al. (2012), Krusche et al. (2018), Krusche and Alperowitz (2014c), Kuhrmann et al. (2013), Lattanze (2016), Lee et al. (2015), Liew (2013), Loksa et al. (2013), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Matthies (2018), Missiroli et al. (2016), Murphy et al. (2017), Nersesian and Spryszynski (2018), Neyem et al. (2014, 2018), Olson and Gibbons (2018), Paasivaara et al. (2013, 2015, 2018), Paiva and Carvalho (2018), Penzenstadler et al. (2013), Pérez-Castillo et al. (2018), Pirker et al. (2016), Portela et al. (2017), Prior et al. (2014), Rodriguez et al. (2015), Rodríguez et al. (2016), Root et al. (2008), Rosca (2018), Rusu and Swenson (2008), Santos et al. (2018), Scharff and Verma (2010), Scott et al. (2016), Smith et al. (2011), Soundararajan et al. (2012), Stapel et al. (2008), Steghöfer et al. (2016), Steghöfer (2018), Venson et al. (2016b) and Wallace et al. (2012)
Gamified learning	Bartholomew (2017), Heikkilä et al. (2016), Hof et al. (2017), Molléri et al. (2018), Pirker et al. (2016) and Rong et al. (2014)
Blended learning	Baldauf et al. (2017), Dow et al. (2013), Murphy et al. (2008), Williams et al. (2015), Wong (2016), Xiao and Miller (2014) and Billingsley and Steel (2014)
Experiential learning	Ahmad et al. (2014)
Other	Caspersen and Kolling (2009), Christensen (2009), Dahotre et al. (2011), Kropp and Meier (2014), Kruchten (2011), Nandi and Mandernach (2016), Paasivaara et al. (2014) and Pérez-Castillo et al. (2018)

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 24
Studies per Stakeholder.

Stakeholder	Studies
Education	Ahmad et al. (2014), Ahmad et al. (2014), Bai et al. (2018a), Baldauf et al. (2017), Bartholomew (2017), Blasquez and Leblanc (2018, 2017), Bosnić et al. (2015), Browning and Sigman (2016), Buffardi (2018, 2017), Buffardi and Edwards (2012), Buffardi et al. (2017a,b), Campbell and Tafliovich (2015), Caspersen and Kolling (2009), Chanin et al. (2018), Chatley and Field (2017), Choudhari and Suman (2015), Christensen (2009), Corral and Fronza (2018), Dahotre et al. (2011), Damian et al. (2012), Davis and Bolen (2016), Delgado et al. (2017), Duvall et al. (2018), Fagerholm et al. (2017), Felker et al. (2012), Fernanda et al. (2018), Fitsilis and Lekatos (2017), Fu et al. (2018), Goto et al. (2014), Heggen and Cody (2018), Heikkilä et al. (2016), Heinonen et al. (2013), Hof et al. (2017), Igaki et al. (2014), Iyengar (2009), Knudson and Radermacher (2011), Kollanus and Isomöttönen (2008), Kropp and Meier (2014), Krusche et al. (2018), Kudikyala and Dulhare (2015), Kuhrmann et al. (2013), Liew (2013), Lynch et al. (2011), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Marques et al. (2018), Matthies (2018), Pirker et al. (2016), Missiroli et al. (2017), Missiroli et al. (2016), Molléri et al. (2018), Murphy et al. (2008, 2017), Neyem et al. (2014, 2018), Olson and Gibbons (2018), Paasivaara et al. (2015, 2014, 2013), Paiva and Carvalho (2018), Pérez-Castillo et al. (2018), Potineni et al. (2013), Rodriguez et al. (2015), Rodríguez et al. (2016), Rodríguez et al. (2018), Santos et al. (2018), Scott et al. (2016), Sievi-Korte et al. (2015), Smith et al. (2011), Soundararajan et al. (2012), de Souza et al. (2015), Stapel et al. (2008), Steghöfer (2018), Steghöfer et al. (2016), Uskov et al. (2016), Villavicencio et al. (2017), Vu et al. (2009), Wallace et al. (2012), Williams et al. (2015), Wong (2016), Xiao and Miller (2014), Palacin-Silva et al. (2017), Prior et al. (2014), Bull et al. (2013), Root et al. (2008), Bull and Whittle (2014), Loksa et al. (2013), Billingsley and Steel (2014), Rusu and Swenson (2008), Honig (2008), Portela et al. (2017), Almeida et al. (2012), Rong et al. (2014), Eddy et al. (2017), Bai et al. (2018b) and Pérez-Castillo et al. (2018)
Education and industry	Bruegge et al. (2015), Brügge and Gluchow (2012), Devadiga (2017), Dow et al. (2013), Kizaki et al. (2014), Kruchten (2011), de Lange et al. (2016), Llopis and Guerrero (2018), Nandi and Mandernach (2016), Nersesian and Spryszynski (2018), Nguyen-Duc et al. (2016), Paasivaara et al. (2018), Pirker et al. (2016), Rico and Sayani (2009), Scharff and Verma (2010), Lee et al. (2015), Lattanze (2016), Kopczyńska et al. (2012), Rosca (2018), Bosnic et al. (2010), Venson et al. (2016b), Dagnino (2014), Huang and Port (2011), Penzenstadler et al. (2013), Chenoweth (2008) and Krusche and Alperowitz (2014c)

Table 25
Studies per contribution type classification.

Contribution type	Studies
Model	Buffardi et al. (2017b), Chatley and Field (2017), Goto et al. (2014), Kuhrmann et al. (2013), Nguyen-Duc et al. (2016), Bull et al. (2013), Huang and Port (2011) and Penzenstadler et al. (2013)
Theory	[ ]
Framework/method	Christensen (2009), Corral and Fronza (2018), Caspersen and Kolling (2009), Chanin et al. (2018), Damian et al. (2012), Duvall et al. (2018), Felker et al. (2012), Fernanda et al. (2018), Kudikyala and Dulhare (2015), de Lange et al. (2016), Llopis and Guerrero (2018), Lynch et al. (2011), Marques et al. (2018), Potineni et al. (2013), Villavicencio et al. (2017), Bai et al. (2018a), Browning and Sigman (2016), Bruegge et al. (2015), Brügge and Gluchow (2012), Lattanze (2016), Rosca (2018), Bosnic et al. (2010) and Venson et al. (2016b)
Survey	Ahmad et al. (2014), Ahmad et al. (2014), Baldauf et al. (2017), Blasquez and Leblanc (2018, 2017), Bosnić et al. (2015), Buffardi (2018, 2017), Buffardi and Edwards (2012), Buffardi et al. (2017a), Choudhari and Suman (2015), Davis and Bolen (2016), Delgado et al. (2017), Devadiga (2017), Dow et al. (2013), Fitsilis and Lekatos (2017), Fu et al. (2018), Heggen and Cody (2018), Heikkilä et al. (2016), Heinonen et al. (2013), Hof et al. (2017), Igaki et al. (2014), Iyengar (2009), Kizaki et al. (2014), Knudson and Radermacher (2011), Kollanus and Isomöttönen (2008), Kropp and Meier (2014), Krusche et al. (2018), Liew (2013), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Matthies (2018), Missiroli et al. (2017), Missiroli et al. (2016), Murphy et al. (2008, 2017), Nandi and Mandernach (2016), Nersesian and Spryszynski (2018), Olson and Gibbons (2018), Paasivaara et al. (2015, 2014, 2013, 2018), Paiva and Carvalho (2018), Palacin-Silva et al. (2017), Pérez-Castillo et al. (2018), Pirker et al. (2016), Rico and Sayani (2009), Rodriguez et al. (2015), Rodríguez et al. (2016), Rodríguez et al. (2018), Santos et al. (2018), Scharff and Verma (2010), Scott et al. (2016), Smith et al. (2011), de Souza et al. (2015), Steghöfer (2018), Steghöfer et al. (2016), Vu et al. (2009), Wallace et al. (2012), Billingsley and Steel (2014), Chenoweth (2008), Honig (2008), Portela et al. (2017), Almeida et al. (2012), Rong et al. (2014), Eddy et al. (2017), Bai et al. (2018b), Pérez-Castillo et al. (2018) and Krusche and Alperowitz (2014c)
Guidelines	Fagerholm et al. (2017), Pirker et al. (2016), Sievi-Korte et al. (2015), Stapel et al. (2008), Uskov et al. (2016), Bull and Whittle (2014) and Dagnino (2014)
Lessons learned	Campbell and Tafliovich (2015), Kruchten (2011), Molléri et al. (2018), Neyem et al. (2014), Soundararajan et al. (2012), Wong (2016), Xiao and Miller (2014), Lee et al. (2015), Prior et al. (2014), Root et al. (2008), Kopczyńska et al. (2012) and Rusu and Swenson (2008)
Advice implications	[ ]
Tool	Bartholomew (2017), Dahotre et al. (2011), Neyem et al. (2018), Williams et al. (2015) and Loksa et al. (2013)

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 26
Studies per research type classification.

Research type	Studies
Validation research	Ahmad et al. (2014), Ahmad et al. (2014), Bai et al. (2018a), Baldauf et al. (2017), Blasquez and Leblanc (2018, 2017), Bosnić et al. (2015), Browning and Sigman (2016), Buffardi (2017), Buffardi and Edwards (2012), Buffardi et al. (2017a,b), Campbell and Tafliovich (2015), Caspersen and Kolling (2009), Chanin et al. (2018), Chatley and Field (2017), Choudhari and Suman (2015), Damian et al. (2012), Davis and Bolen (2016), Delgado et al. (2017), Duvall et al. (2018), Fernanda et al. (2018), Fitsilis and Lekatos (2017), Goto et al. (2014), Heggen and Cody (2018), Heikkilä et al. (2016), Heinonen et al. (2013), Hof et al. (2017), Igaki et al. (2014), Iyengar (2009), Knudson and Radermacher (2011), Kollanus and Isomöttönen (2008), Kropp and Meier (2014), Krusche et al. (2018), Kudikyala and Dulhare (2015), Kuhrmann et al. (2013), Liew (2013), Llopis and Guerrero (2018), Lynch et al. (2011), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Marques et al. (2018), Matthies (2018), Olson and Gibbons (2018), Paasivaara et al. (2015, 2014, 2013, 2018), Paiva and Carvalho (2018), Palacin-Silva et al. (2017), Pérez-Castillo et al. (2018), Pirker et al. (2016), Potineni et al. (2013), Rodríguez et al. (2016), Santos et al. (2018), Scott et al. (2016), Smith et al. (2011), de Souza et al. (2015), Steghöfer (2018), Steghöfer et al. (2016), Villavicencio et al. (2017), Vu et al. (2009), Lee et al. (2015), Prior et al. (2014), Bull et al. (2013), Loksa et al. (2013), Rosca (2018), Billingsley and Steel (2014), Chenoweth (2008), Honig (2008), Eddy et al. (2017), Bai et al. (2018b), Pérez-Castillo et al. (2018), Krusche and Alperowitz (2014c), Missiroli et al. (2017), Missiroli et al. (2016), Molléri et al. (2018), Murphy et al. (2008, 2017), Nandi and Mandernach (2016), Nersesian and Spryszynski (2018), Neyem et al. (2014, 2018) and Rodriguez et al. (2015)
Evaluation research	de Lange et al. (2016), Bruegge et al. (2015), Buffardi (2018), Devadiga (2017), Dow et al. (2013), Kizaki et al. (2014), Nguyen-Duc et al. (2016), Rico and Sayani (2009), Uskov et al. (2016), Wallace et al. (2012), Huang and Port (2011), Venson et al. (2016b), Rodríguez et al. (2018) and Scharff and Verma (2010)
Solution proposal	Bartholomew (2017), Brügge and Gluchow (2012), Corral and Fronza (2018), Dahotre et al. (2011), Felker et al. (2012), Williams et al. (2015), Root et al. (2008) and Lattanze (2016)
Philosophical paper	[ ]
Opinion paper	[ ]
Experience paper	Christensen (2009), Fagerholm et al. (2017), Fu et al. (2018), Kruchten (2011), Pirker et al. (2016), Sievi-Korte et al. (2015), Soundararajan et al. (2012), Stapel et al. (2008), Wong (2016), Xiao and Miller (2014), Bull and Whittle (2014), Kopczyńska et al. (2012), Dagnino (2014), Portela et al. (2017), Bosnic et al. (2010), Almeida et al. (2012), Rong et al. (2014), Rusu and Swenson (2008) and Penzenstadler et al. (2013)

Table 27
Venues.

Venues	Studies
Journal	Ahmad et al. (2014), Bartholomew (2017), Browning and Sigman (2016), Bruegge et al. (2015), Buffardi et al. (2017b), Caspersen and Kolling (2009), Choudhari and Suman (2015), Dahotre et al. (2011), Krusche et al. (2018), Marques et al. (2018), Neyem et al. (2018), Rodríguez et al. (2016), Scott et al. (2016), Rodriguez et al. (2015) and Bull and Whittle (2014)
Conference	Chanin et al. (2018), Davis and Bolen (2016), Sievi-Korte et al. (2015), Ahmad et al. (2014), Bai et al. (2018a), Baldauf et al. (2017), Blasquez and Leblanc (2018, 2017), Buffardi (2017), Buffardi and Edwards (2012), Buffardi et al. (2017a), Campbell and Tafliovich (2015), Chatley and Field (2017), Christensen (2009), Corral and Fronza (2018), Delgado et al. (2017), Devadiga (2017), Dow et al. (2013), Duvall et al. (2018), Fagerholm et al. (2017), Felker et al. (2012), Fernanda et al. (2018), Fitsilis and Lekatos (2017), Fu et al. (2018), Goto et al. (2014), Heikkilä et al. (2016), Heinonen et al. (2013), Hof et al. (2017), Igaki et al. (2014), Iyengar (2009), Kizaki et al. (2014), Knudson and Radermacher (2011), Kollanus and Isomöttönen (2008), Kropp and Meier (2014), Kruchten (2011), Kudikyala and Dulhare (2015), Kuhrmann et al. (2013), Liew (2013), Llopis and Guerrero (2018), Lynch et al. (2011), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Pirker et al. (2016), Missiroli et al. (2016), Molléri et al. (2018), Murphy et al. (2008, 2017), Nandi and Mandernach (2016), Nersesian and Spryszynski (2018), Neyem et al. (2014), Nguyen-Duc et al. (2016), Olson and Gibbons (2018), Paasivaara et al. (2015, 2014, 2013, 2018), Paiva and Carvalho (2018), Palacin-Silva et al. (2017), Pérez-Castillo et al. (2018), Pirker et al. (2016), Potineni et al. (2013), Rico and Sayani (2009), Rodríguez et al. (2018), Santos et al. (2018), Soundararajan et al. (2012), de Souza et al. (2015), Stapel et al. (2008), Steghöfer (2018), Steghöfer et al. (2016), Uskov et al. (2016), Villavicencio et al. (2017), Vu et al. (2009), Wallace et al. (2012), Williams et al. (2015), Wong (2016), Xiao and Miller (2014), Lee et al. (2015), Prior et al. (2014), Bull et al. (2013), Root et al. (2008), Lattanze (2016), Loksa et al. (2013), Rosca (2018), Billingsley and Steel (2014), Chenoweth (2008), Honig (2008), Dagnino (2014), Portela et al. (2017), Huang and Port (2011), Bosnic et al. (2010), Almeida et al. (2012), Rong et al. (2014), Eddy et al. (2017), Rusu and Swenson (2008), Venson et al. (2016b), Bai et al. (2018b), Pérez-Castillo et al. (2018), Penzenstadler et al. (2013), Krusche and Alperowitz (2014c) and de Lange et al. (2016)
Workshop	Bosnić et al. (2015), Brügge and Gluchow (2012), Buffardi (2018), Damian et al. (2012), Heggen and Cody (2018), Matthies (2018), Missiroli et al. (2017), Scharff and Verma (2010), Smith et al. (2011) and Kopczyńska et al. (2012)

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.

Table 28
Database and Manual Searches.

Search type	Studies
Database search	Ahmad et al. (2014), Baldauf et al. (2017), Bartholomew (2017), Blasquez and Leblanc (2018, 2017), Bosnić et al. (2015), Browning and Sigman (2016), Bruegge et al. (2015), Brügge and Gluchow (2012), Buffardi (2018, 2017), Buffardi and Edwards (2012), Buffardi et al. (2017a,b), Campbell and Tafliovich (2015), Caspersen and Kolling (2009), Chanin et al. (2018), Choudhari and Suman (2015), Christensen (2009), Corral and Fronza (2018), Dahotre et al. (2011), Damian et al. (2012), Davis and Bolen (2016), Delgado et al. (2017), Dow et al. (2013), Duvall et al. (2018), Fagerholm et al. (2017), Felker et al. (2012), Fitsilis and Lekatos (2017), Fu et al. (2018), Goto et al. (2014), Heggen and Cody (2018), Heikkilä et al. (2016), Heinonen et al. (2013), Hof et al. (2017), Iyengar (2009), Kizaki et al. (2014), Knudson and Radermacher (2011), Kollanus and Isomöttönen (2008), Kropp and Meier (2014), Kruchten (2011), Krusche et al. (2018), Kudikyala and Dulhare (2015), Kuhrmann et al. (2013), de Lange et al. (2016), Liew (2013), Llopis and Guerrero (2018), Lyra et al. (2018), Ma et al. (2018), Mahnic and Rozanc (2012), Marques et al. (2018), Matthies (2018), Pirker et al. (2016), Missiroli et al. (2017), Missiroli et al. (2016), Molléri et al. (2018), Murphy et al. (2008, 2017), Nandi and Mandernach (2016), Nersesian and Spryszynski (2018), Neyem et al. (2014, 2018), Nguyen-Duc et al. (2016), Olson and Gibbons (2018), Paasivaara et al. (2015, 2014, 2013), Paiva and Carvalho (2018), Pirker et al. (2016), Potineni et al. (2013), Rico and Sayani (2009), Rodriguez et al. (2015), Rodríguez et al. (2016), Scharff and Verma (2010), Scott et al. (2016), Smith et al. (2011), Soundararajan et al. (2012), Stapel et al. (2008), Steghöfer (2018), Steghöfer et al. (2016), Uskov et al. (2016), Villavicencio et al. (2017), Vu et al. (2009), Wallace et al. (2012), Williams et al. (2015), Wong (2016), Xiao and Miller (2014), Lee et al. (2015), Prior et al. (2014), Bull et al. (2013), Root et al. (2008), Bull and Whittle (2014), Lattanze (2016), Kopczyńska et al. (2012), Loksa et al. (2013), Rosca (2018) and Billingsley and Steel (2014)
Manual	Bai et al. (2018a), Chatley and Field (2017), Paasivaara et al. (2018), Santos et al. (2018), Igaki et al. (2014), Devadiga (2017), Palacin-Silva et al. (2017), Sievi-Korte et al. (2015), de Souza et al. (2015), Lynch et al. (2011), Pérez-Castillo et al. (2018), Rodríguez et al. (2018), Ahmad et al. (2014), Fernanda et al. (2018), Chenoweth (2008), Honig (2008), Dagnino (2014), Portela et al. (2017), Huang and Port (2011), Bosnic et al. (2010), Almeida et al. (2012), Rong et al. (2014), Eddy et al. (2017), Rusu and Swenson (2008), Venson et al. (2016b), Bai et al. (2018b), Pérez-Castillo et al. (2018), Penzenstadler et al. (2013) and Krusche and Alperowitz (2014c)

References

A. f. C. M. A. Joint Task Force on Computing Curricula, I. C. Society, 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, New York, NY, USA.

Ahmad, M.O., Liukkunen, K., Markkula, J., 2014. Student perceptions and attitudes towards the software factory as a learning environment. In: 2014 IEEE Global Engineering Education Conference (EDUCON). pp. 422–428.

Ahmad, M., Markkula, J., Oivo, M., 2014. Kanban for software engineering teaching in a software factory learning environment. World Trans. Eng. Technol. Educ. 12 (3), 338–343, cited By 4.

Almeida, E., Dali, L., Faulk, S., Lima, C., Rui, Z., Weiss, D., Ying, J., Young, M., Yu, L., 2012. Teaching globally distributed software development: An experience report. In: 2012 IEEE 25th Conference on Software Engineering Education and Training. IEEE, pp. 105–109.

Almi, N.E.A.M., Rahman, N.A., Purusothaman, D., Sulaiman, S., 2011. Software engineering education: The gap between industry's requirements and graduates' readiness. In: 2011 IEEE Symposium on Computers Informatics. pp. 542–547.

Almi, N.E.A.M., Rahman, N.A., Purusothaman, D., Sulaiman, S., 2011. Software engineering education: The gap between industry's requirements and graduates' readiness. In: Computers & Informatics (ISCI), 2011 IEEE Symposium on. IEEE, pp. 542–547.

Anon, 2019a. Scimago. URL https://www.scimagojr.com/, accessed 25 March 2019.

Anon, 2019b. Thomson Reuters. URL https://jcr.clarivate.com/JCRLandingPageAction.action?, accessed 25 March 2019.

Anslow, C., Maurer, F., 2015. An experience Report at Teaching a Group based Agile Software Development Project Course. In: Proceedings of the 46th ACM Technical Symposium on Computer Science Education. In: SIGCSE '15, ACM, New York, NY, USA, pp. 500–505.

Bai, X., Li, M., Pei, D., Li, S., Ye, D., 2018a. Continuous delivery of Personalized Assessment and Feedback in Agile Software Engineering Projects. In: Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training. In: ICSE-SEET '18, ACM, New York, NY, USA, pp. 58–67.

Bai, X., Pei, D., Li, M., Li, S., 2018b. The DevOps Lab Platform for Managing Diversified Projects in Educating Agile Software Engineering. In: 2018 IEEE Frontiers in Education Conference (FIE). IEEE, pp. 1–5.

Baldauf, M., Brandner, A., Wimmer, C., 2017. Mobile and Gamified Blended Learning for Language teaching: Studying requirements and acceptance by students, parents and teachers in the wild. In: Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia. In: MUM '17, ACM, New York, NY, USA, pp. 13–24.

Bartholomew, K.W., 2017. Agile Gamification: Activities and Techniques to Create a Winning Learning Environment. J. Comput. Sci. Coll. 33 (2), 57–58.

Bass, M., 2016. Software Engineering Education in the New World: What needs to change? In: 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET). pp. 213–221.

Bastarrica, M.C., Perovich, D., Samary, M.M., 2017. What can students get from a Software Engineering Capstone Course? In: 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET). pp. 137–145.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al., 2001. Manifesto for agile software development.

Beckman, K., Coulter, N., Khajenoori, S., Mead, N.R., 1997. Collaborations: closing the industry-academia gap. IEEE Softw. 14 (6), 49–57.

Beecham, S., Clear, T., Barr, J., Daniels, M., Oudshoorn, M., Noll, J., 2017a. Preparing Tomorrow's Software Engineers for work in a global Environment. IEEE Softw. 34 (1), 9–12.

Beecham, S., Clear, T., Noll, J., 2017b. Do we teach the right thing?: a comparison of global software engineering education and practice. In: Proceedings of the 12th International Conference on Global Software Engineering. IEEE Press, pp. 11–20.

Begel, A., Simon, B., 2008a. Novice software developers, all over again. In: Proceedings of the Fourth International Workshop on Computing Education Research. pp. 3–14.

Begel, A., Simon, B., 2008b. Struggles of new college graduates in their first software development job. In: Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education. pp. 226–230.

Bezemer, C.-P., Eismann, S., Ferme, V., Grohmann, J., Heinrich, R., Jamshidi, P., Shang, W., van Hoorn, A., Villavicencio, M., Walter, J., Willnecker, F., 2019. How is performance Addressed in DevOps? In: Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. In: ICPE '19, ACM, New York, NY, USA, pp. 45–50.

Billingsley, W., Steel, J.R., 2014. Towards a supercollaborative software engineering MOOC. In: Companion Proceedings of the 36th International Conference on Software Engineering. pp. 283–286.

Blasquez, I., Leblanc, H., 2017. Specification by example for Educational Purposes. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education. In: ITiCSE '17, ACM, New York, NY, USA, pp. 212–217.

Blasquez, I., Leblanc, H., 2018. Experience in Learning Test-driven Development: Space Invaders Project-driven. In: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education. In: ITiCSE 2018, ACM, New York, NY, USA, pp. 111–116.

Boehm, B., 2006. A view of 20th and 21st Century Software Engineering. In: Proceedings of the 28th International Conference on Software Engineering. In: ICSE '06, ACM, New York, NY, USA, pp. 12–29.

Bolinger, J., Yackovich, K., Ramnath, R., Ramanathan, J., Soundarajan, N., 2010. From student to teacher: Transforming industry sponsored student Projects into Relevant, engaging, and Practical Curricular Materials. In: 2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments. pp. 1–21.

Bollin, A., Reçi, E., Szabó, C., Szabóová, V., Siebenhofer, R., 2018. Applying a maturity model during a software engineering course–How planning and task-solving processes influence the course performance. J. Syst. Softw. 144, 397–408.

Bosch, J., Holmström Olsson, H., Björk, J., Ljungblad, J., 2013a. The early stage Software Startup development model: A Framework for Operationalizing lean principles in Software Startups. In: Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J. (Eds.), Lean Enterprise Software and Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–15.

Bosch, J., Olsson, H.H., Björk, J., Ljungblad, J., 2013b. The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: International Conference on Lean Enterprise Software and Systems. Springer, pp. 1–15.

Bosnic, I., Cavrak, I., Žagar, M., Land, R., Crnkovic, I., 2010. Customers' role in teaching distributed software development. In: 2010 23rd IEEE Conference on Software Engineering Education and Training. IEEE, pp. 73–80.

Bosnić, I., Ciccozzi, F., Čavrak, I., Di Nitto, E., Feljan, J., Mirandola, R., 2015. Introducing SCRUM into a Distributed Software Development Course. In: Proceedings of the 2015 European Conference on Software Architecture Workshops. In: ECSAW '15, ACM, New York, NY, USA, pp. 34:1–34:8.

Browning, C., Sigman, S., 2016. Pedal: a pedagogical software development process designed for student success. J. Comput. Sci. Coll. 31 (5), 171–178.

Bruegge, B., Krusche, S., Alperowitz, L., 2015. Software Engineering Project Courses with Industrial Clients. Trans. Comput. Educ. 15 (4), 17:1–17:31.

Brügge, B., Gluchow, M., 2012. Towards production ready software in project courses with real clients. In: 2012 First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex). IEEE, pp. 5–8.

Buffardi, K., 2017. Comparing remote and Co-located Interaction in Free and Open Source Software Engineering Projects. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education. ACM, pp. 22–27.

Buffardi, K., 2018. Tech startup Learning Activities: A formative Evaluation. In: Proceedings of the 2Nd International Workshop on Software Engineering Education for Millennials. In: SEEM '18, ACM, New York, NY, USA, pp. 24–31.

Buffardi, K., Edwards, S.H., 2012. Exploring Influences on Student Adherence to test-driven development. In: Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education. In: ITiCSE '12, ACM, New York, NY, USA, pp. 105–110.

Buffardi, K., Robb, C., Rahn, D., 2017a. Learning Agile with Tech Startup Software engineering Projects. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education. In: ITiCSE '17, ACM, New York, NY, USA, pp. 28–33.

Buffardi, K., Robb, C., Rahn, D., 2017b. Tech startups: Realistic software engineering projects with interdisciplinary collaboration. J. Comput. Sci. Coll. 32 (4), 93–98.

Bull, C.N., Whittle, J., 2014. Supporting reflective practice in software engineering education through a studio-based approach. IEEE Softw. 31 (4), 44–50.

Bull, C.N., Whittle, J., Cruickshank, L., 2013. Studios in software engineering education: towards an evaluable model. In: 2013 35th International Conference on Software Engineering (ICSE). IEEE, pp. 1063–1072.

Campbell, J., Tafliovich, A., 2015. An Experience Report: Using mobile development to Teach Software Design. In: Proceedings of the 46th ACM Technical Symposium on Computer Science Education. In: SIGCSE '15, ACM, New York, NY, USA, pp. 506–511.

Carter, L., 2011. Ideas for adding soft skills education to service learning and capstone courses for computer science students. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education. pp. 517–522.

Caspersen, M.E., Kolling, M., 2009. STREAM: A first Programming Process. Trans. Comput. Educ. 9 (1), 4:1–4:29.

Chanin, R., Sales, A., Pompermaier, L., Prikladnicki, R., 2018. 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. In: ITiCSE 2018, ACM, New York, NY, USA, pp. 266–271.

Charmaz, K., Belgrave, L.L., 2007. Grounded theory. In: The Blackwell Encyclopedia of Sociology. Wiley Online Library.

Chatley, R., Field, T., 2017. Lean learning - applying lean Techniques to Improve Software Engineering Education. In: 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET). pp. 117–126.

Chenoweth, S., 2008. Undergraduate software engineering students in startup businesses. In: 2008 21st Conference on Software Engineering Education and Training. IEEE, pp. 118–125.

Choudhari, J., Suman, U., 2015. An empirical evaluation of iterative maintenance life cycle using xp. ACM SIGSOFT Softw. Eng. Notes 40 (2), 1–14.

Christensen, H.B., 2009. A story-telling approach for a Software Engineering Course Design. In: Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education. In: ITiCSE '09, ACM, New York, NY, USA, pp. 60–64.

Clear, T., Beecham, S., 2019. Global Software Engineering Education practice continuum special issue of the ACM transactions on computing education. ACM Trans. Comput. Educ. 19 (2), 7:1–7:8.

Clear, T., Beecham, S., Barr, J., Daniels, M., McDermott, R., Oudshoorn, M., Savickaite, A., Noll, J., 2015. Challenges and recommendations for the design and Conduct of Global Software Engineering Courses: A systematic review. In: Proceedings of the 2015 ITiCSE on Working Group Reports. In: ITICSE-WGR '15, ACM, New York, NY, USA, pp. 1–39.

Clear, T., Beecham, S., Barr, J., Daniels, M., Oudshoorn, M., Noll, J., 2016. Developments in Global Software Engineering Education. In: 2016 IEEE Frontiers in Education Conference (FIE). pp. 1–4.

Clear, A., Parrish, A.S., Impagliazzo, J., Zhang, M., 2019. Computing Curricula 2020: Introduction and Community Engagement. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education. In: SIGCSE '19, ACM, New York, NY, USA, pp. 653–654.

Corral, L., Fronza, I., 2018. Design thinking and Agile Practices for Software Engineering: An Opportunity for Innovation. In: Proceedings of the 19th Annual SIG Conference on Information Technology Education. In: SIGITE '18, ACM, New York, NY, USA, pp. 26–31.

Dagnino, A., 2014. Increasing the effectiveness of teaching software engineering: A university and industry partnership. In: 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 49–54.

Dahotre, A., Krishnamoorthy, V., Corley, M., Scaffidi, C., 2011. Using Intelligent Tutors to Enhance Student Learning of application Programming Interfaces. J. Comput. Sci. Coll. 27 (1), 195–201.

Damian, D., Lassenius, C., Paasivaara, M., Borici, A., Schröter, A., 2012. Teaching a Globally Distributed Project course using Scrum Practices. In: Proceedings of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development. In: CTGDSD '12, IEEE Press, Piscataway, NJ, USA, pp. 30–34.

Davis, C.E., Bolen, R.E., 2016. Implementing Lean LaunchPad methodology into an engineering professional development course. In: 2016 IEEE Frontiers in Education Conference (FIE). pp. 1–6.

De Freitas, S.I., Morgan, J., Gibson, D., 2015. Will MOOCs transform learning and teaching in higher education? Engagement and course retention in online learning provision. Br. J. Educ. Technol. 46 (3), 455–471.

Delgado, D., Velasco, A., Aponte, J., Marcus, A., 2017. Evolving a Project-based Software Engineering Course: A case study. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T). pp. 77–86.

Devadiga, N.M., 2017. Software engineering education: Converging with the startup industry. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 192–196.

Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B., 2012. A decade of Agile Methodologies. J. Syst. Softw. 85 (6), 1213–1221.

Diniz, G.C., Silva, M.A.G., Gerosa, M.A., Steinmacher, I., 2017. Using Gamification to Orient and Motivate Students to contribute to OSS projects. In: 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). pp. 36–42.

Dow, S., Gerber, E., Wong, A., 2013. A pilot study of using crowds in the classroom. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 227–236.

Duvall, S., Hutchings, D.R., Duvall, R.C., 2018. Scrumage: A method for Incorporating Multiple, simultaneous Pedagogical styles in the Classroom. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. In: SIGCSE '18, ACM, New York, NY, USA, pp. 928–933.

Easterbrook, S., Singer, J., Storey, M.-A., Damian, D., 2008. Selecting empirical methods for software engineering research. In: Guide to Advanced Empirical Software Engineering. Springer, pp. 285–311.

Eddy, B.P., Wilde, N., Cooper, N.A., Mishra, B., Gamboa, V.S., Shah, K.M., Deleon, A.M., Shields, N.A., 2017. A pilot study on introducing continuous integration and delivery into undergraduate software engineering courses. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 47–56.

Elberzhager, F., Münch, J., Nha, V.T.N., 2012. A systematic mapping study on the combination of static and dynamic quality assurance techniques. Inf. Softw. Technol. 54 (1), 1–15.

Fagerholm, F., Hellas, A., Luukkainen, M., Kyllönen, K., Yaman, S., Mäenpää, H., 2017. Patterns for designing and implementing an Environment for Software Start-up Education. In: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp. 133–140.

Felker, C., Slamova, R., Davis, J., 2012. Integrating UX with Scrum in an undergraduate Software Development Project. In: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education. In: SIGCSE '12, ACM, New York, NY, USA, pp. 301–306.

Fernanda, S., Manuel, S., Germania, R., Samanta, C., Danilo, J., Patricio, A., 2018. Agile methodologies applied in teaching-learning process in engineering: A case of study. In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 1201–1207.

Fitsilis, P., Lekatos, A., 2017. Teaching Software Project Management using Agile Paradigm. In: Proceedings of the 21st Pan-Hellenic Conference on Informatics. In: PCI 2017, ACM, New York, NY, USA, pp. 47:1–47:6.

Fitzgerald, B., Stol, K.J., 2017. Continuous software engineering: A roadmap and agenda. J. Syst. Softw. 123, 176–189.

Fu, Y., Reina, L.P., Brockmann, P., 2018. Teaching Global Software Engineering: Experience report comparing distributed, virtual collaborative courses at the Bachelor's and Master's Degree Levels. In: Proceedings of the 3rd European Conference of Software Engineering Education. In: ECSEE'18, ACM, New York, NY, USA, pp. 34–38.

Garousi, V., Petersen, K., Ozkan, B., 2016. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. Inf. Softw. Technol. 79, 106–127.

Goto, T., Tsuchida, K., Nishino, T., 2014. EPISODE: An extreme programming method for Innovative Software based on Systems Design. In: 2014 IIAI 3rd International Conference on Advanced Applied Informatics. pp. 780–784.

Heggen, S., Cody, M., 2018. Hiring Millennial Students as Software Engineers: A study in developing self-confidence and Marketable Skills. In: 2018 IEEE/ACM International Workshop on Software Engineering Education for Millennials (SEEM). pp. 32–39.

Heikkilä, V.T., Paasivaara, M., Lassenius, C., 2016. Teaching university students kanban with a collaborative board game. In: Proceedings of the 38th International Conference on Software Engineering Companion. In: ICSE '16, ACM, New York, NY, USA, pp. 471–480.

Heinonen, K., Hirvikoski, K., Luukkainen, M., Vihavainen, A., 2013. Learning agile Software Engineering Practices using coding Dojo. In: Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education. In: SIGITE '13, ACM, New York, NY, USA, pp. 97–102.

Hof, S., Kropp, M., Landolt, M., 2017. Use of Gamification to Teach Agile values and Collaboration: A multi-week Scrum Simulation Project in an undergraduate Software Engineering Course. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education. In: ITiCSE '17, ACM, New York, NY, USA, pp. 323–328.

Holmes, R., Allen, M., Craig, M., 2018. Dimensions of Experientialism for Software Engineering Education. In: Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training. In: ICSE-SEET '18, ACM, New York, NY, USA, pp. 31–39.

Honig, W.L., 2008. Teaching successful" Real-World" Software Engineering to the" Net" Generation: Process and Quality Win!. In: 2008 21st Conference on Software Engineering Education and Training. IEEE, pp. 25–32.

Huang, L., Port, D., 2011. Relevance and alignment of Real-Client Real-Project courses via technology transfer. In: 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 189–198.

Igaki, H., Fukuyasu, N., Saiki, S., Matsumoto, S., Kusumoto, S., 2014. Quantitative assessment with using Ticket Driven Development for Teaching Scrum Framework. In: Companion Proceedings of the 36th International Conference on Software Engineering. In: ICSE Companion 2014, ACM, New York, NY, USA, pp. 372–381.

Impagliazzo, J., Clear, A., Alrumaih, H., 2018. Developing an overview of Computing/Engineering Curricula via the CC2020 Project. In: 2018 IEEE World Engineering Education Conference (EDUNINE). pp. 1–4.

Iyengar, S.R., 2009. Teaching enterprise Software Development in undergraduate Curriculum. In: Proceedings of the 10th ACM Conference on SIG-Information Technology Education. In: SIGITE '09, ACM, New York, NY, USA, pp. 29–32.

Jaakkola, H., Henno, J., Rudas, I.J., 2006. IT curriculum as a Complex Emerging Process. In: 2006 IEEE International Conference on Computational Cybernetics. pp. 1–5.

Järvi, A., Taajamaa, V., Hyrynsalmi, S., 2015. Lean software startup–an experience report from an entrepreneurial software business course. In: International Conference of Software Business. Springer, pp. 230–244.

Kitchenham, B., Charters, S., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Citeseer.

Kizaki, S., Tahara, Y., Ohsuga, A., 2014. Software Development PBL Focusing on Communication using Scrum. In: 2014 IIAI 3rd International Conference on Advanced Applied Informatics. pp. 662–669.

Knudson, D., Radermacher, A., 2011. Updating CS capstone projects to incorporate new agile methodologies used in industry. In: 2011 24th IEEE-CS Conference on Software Engineering Education and Training. pp. 444–448.

Kollanus, S., Isomöttönen, V., 2008. Test-driven development in education: experiences with critical viewpoints. ACM SIGCSE Bull. 40 (3), 124–127.

Kopczyńska, S., Nawrocki, J., Ochodek, M., 2012. Software development studio-Bringing industrial environment to a classroom. In: 2012 First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex). IEEE, pp. 13–16.

Kropp, M., Meier, A., 2014. New sustainable teaching approaches in software engineering education. In: 2014 IEEE Global Engineering Education Conference (EDUCON). pp. 1019–1022.

Kropp, M., Meier, A., 2016. Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight. In: 2016 IEEE Global Engineering Education Conference (EDUCON). pp. 1003–1011.

Kruchten, P., 2011. Experience teaching software project management in both industrial and academic settings. In: 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 199–208.

Kruchten, P., 2011. Experience teaching software project management in both industrial and academic settings. In: 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE T). pp. 199–208.

Krusche, S., Alperowitz, L., 2014a. Introduction of Continuous Delivery in multi-customer project courses. In: Companion Proceedings of the 36th International Conference on Software Engineering. In: ICSE Companion 2014, ACM, New York, NY, USA, pp. 335–343.

Krusche, S., Alperowitz, L., 2014b. Introduction of continuous delivery in multi-customer project courses. In: Companion Proceedings of the 36th International Conference on Software Engineering. In: ICSE Companion 2014, ACM, New York, NY, USA, pp. 335–343.

Krusche, S., Alperowitz, L., 2014c. Introduction of continuous delivery in multi-customer project courses. In: Companion Proceedings of the 36th International Conference on Software Engineering. pp. 335–343.

Krusche, S., Dzvonyar, D., Xu, H., Bruegge, B., 2018. Software Theater - Teaching Demo-Oriented Prototyping. ACM Trans. Comput. Educ. 18 (2), 10:1–10:30.

Kuang, L.-Q., Han, X., 2012. The research of software engineering curriculum reform. Physics Procedia 33, 1762–1767.

Kudikyala, U.K., Dulhare, U.N., 2015. Using Scrum and Wikis to manage student major projects. In: 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE). pp. 15–20.

Kuhrmann, M., Fernández, D.M., Münch, J., 2013. Teaching Software Process Modeling. In: Proceedings of the 2013 International Conference on Software Engineering. In: ICSE '13, IEEE Press, Piscataway, NJ, USA, pp. 1138–1147.

de Lange, P., Nicolaescu, P., Klamma, R., Koren, I., 2016. Devops use for Rapid Training of Agile Practices within Undergraduate and startup communities. In: European Conference on Technology Enhanced Learning. Springer, pp. 570–574.

Lattanze, A.J., 2016. Practice based studio. In: 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET). IEEE, pp. 1–7.

Lee, J., Kotonya, G., Whittle, J., Bull, C., 2015. Software design studio: a practical example. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Vol. 2. IEEE, pp. 389–397.

Lethbridge, T.C., Diaz-Herrera, J., LeBlanc, R.J.J., Thompson, J.B., 2007. Improving software practice through education: Challenges and future trends. In: Future of Software Engineering (FOSE '07). pp. 12–28.

Liew, C.W., 2013. Benefits of having students develop software for other departments. In: Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education. ACM, p. 348.

Llopis, F., Guerrero, F.G., 2018. Introducing competitiveness and industry involvement as learning tools. In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 298–307.

Loksa, D., Mangano, N., LaToza, T.D., van der Hoek, A., 2013. Enabling a classroom design studio with a collaborative sketch design tool. In: 2013 35th International Conference on Software Engineering (ICSE). IEEE, pp. 1073–1082.

Lynch, T.D., Herold, M., Bolinger, J., Deshpande, S., Bihari, T., Ramanathan, J., Ramnath, R., 2011. An agile boot camp: Using a LEGO®-based active game to ground agile development principles. In: 2011 Frontiers in Education Conference (FIE). F1H–1–F1H–6.

Lyra, K.T., Alves, M.L., Silva, F.H.C., Souza, K., Isotani, S., 2018. An Agile Project Management experience: Points of view of Graduate Students. In: Proceedings of the XXXII Brazilian Symposium on Software Engineering. In: SBES '18, ACM, New York, NY, USA, pp. 240–249.

Ma, K., Yang, B., Zhou, J., Lin, Y., Zhang, K., Yu, Z., 2018. Outcome-based school-enterprise cooperative software engineering training. In: Proceedings of ACM Turing Celebration Conference-China. ACM, pp. 15–20.

Mahnic, V., Rozanc, I., 2012. Students' perceptions of Scrum practices. In: 2012 Proceedings of the 35th International Convention MIPRO. IEEE, pp. 1178–1183.

Marques, M., Ochoa, S.F., Bastarrica, M.C., Gutierrez, F.J., 2018. Enhancing the student learning experience in software engineering project courses. IEEE Trans. Educ. 61 (1), 63–73, month=Feb.

Marques, M.R., Quispe, A., Ochoa, S.F., 2014a. A systematic mapping study on practical approaches to teaching software engineering. In: 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. pp. 1–8.

Marques, M.R., Quispe, A., Ochoa, S.F., 2014b. A systematic mapping study on practical approaches to teaching software engineering. In: 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. IEEE, pp. 1–8.

Matthies, C., 2018. Scrum2Kanban: Integrating Kanban and scrum in a University Software Engineering Capstone Course. In: Proceedings of the 2Nd International Workshop on Software Engineering Education for Millennialsa. In: SEEM '18, ACM, New York, NY, USA, pp. 48–55.

Meier, A., Kropp, M., Perellano, G., 2016. Experience report of teaching agile collaboration and values: Agile software development in large student teams. In: 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET). pp. 76–80.

Missiroli, M., Russo, D., Ciancarini, P., 2016. Learning Agile Software Development in High School: An investigation. In: Proceedings of the 38th International Conference on Software Engineering Companion. In: ICSE '16, ACM, New York, NY, USA, pp. 293–302.

Missiroli, M., Russo, D., Ciancarini, P., 2017. Agile for Millennials: A Comparative Study. In: 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM). pp. 47–53.

Molléri, J.S., Gonzalez-Huerta, J., Henningsson, K., 2018. A Legacy Game for Project Management in Software Engineering Courses. In: Proceedings of the 3rd European Conference of Software Engineering Education. In: ECSEE'18, ACM, New York, NY, USA, pp. 72–76.

Monasor, M.J., Vizcaino, A., Piattini, M., Caballero, I., 2010. Preparing students and engineers for global Software Development: A systematic review. In: 2010 5th IEEE International Conference on Global Software Engineering. pp. 177–186.

Müller, S., Bergande, B., Brune, P., 2018. Robot Tutoring: On the feasibility of using Cognitive Systems as Tutors in Introductory programming education: A teaching experiment. In: Proceedings of the 3rd European Conference of Software Engineering Education. ACM, pp. 45–49.

Müller, S., Bergande, B., Brune, P., 2018. Robot tutoring: On the Feasibility of using Cognitive Systems as Tutors in Introductory Programming Education: A teaching experiment. In: Proceedings of the 3rd European Conference of Software Engineering Education. In: ECSEE'18, ACM, New York, NY, USA, pp. 45–49.

Murphy, C., Phung, D., Kaiser, G., 2008. A distance learning approach to teaching extreme programming. ACM SIGCSE Bull. 40 (3), 199–203.

Murphy, C., Sheth, S., Morton, S., 2017. A two-course sequence of real projects for real customers. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM, pp. 417–422.

Nandi, A., Mandernach, M., 2016. Hackathons as an Informal Learning Platform. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. In: SIGCSE '16, ACM, New York, NY, USA, pp. 346–351.

Nersesian, E., Spryszynski, A., 2018. Incorporating workplace Structure in a Classroom Setting. In: Proceedings of the 19th Annual SIG Conference on Information Technology Education. In: SIGITE '18, ACM, New York, NY, USA, p. 190.

Neyem, A., Benedetto, J.I., Chacon, A.F., 2014. Improving Software Engineering Education through an Empirical approach: Lessons learned from Capstone Teaching Experiences. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education. In: SIGCSE '14, ACM, New York, NY, USA, pp. 391–396.

Neyem, A., Diaz-Mosquera, J., Benedetto, J.I., 2018. A cloud-based mobile system to improve Project Management Skills in Software Engineering Capstone Courses. Mob. Inf. Syst. 2018.

Nguyen-Duc, A., Shah, S.M.A., Ambrahamsson, P., 2016. Towards an Early Stage Software Startups Evolution Model. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). pp. 120–127.

Olson, B., Gibbons, T., 2018. Applying an Agile approach to improve the Student Experience and Performance in a Capstone Project. In: Proceedings of the 19th Annual SIG Conference on Information Technology Education. In: SIGITE '18, ACM, New York, NY, USA, p. 87.

Paasivaara, M., Blincoe, K., Lassenius, C., Damian, D., Sheoran, J., Harrison, F., Chhabra, P., Yussuf, A., Isotalo, V., 2015. Learning global Agile Software Engineering Using Same-Site and cross-site teams. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Vol. 2. pp. 285–294.

Paasivaara, M., Heikkilä, V., Lassenius, C., Toivola, T., 2014. Teaching students Scrum using LEGO Blocks. In: Companion Proceedings of the 36th International Conference on Software Engineering. In: ICSE Companion 2014, ACM, New York, NY, USA, pp. 382–391.

Paasivaara, M., Lassenius, C., Damian, D., Räty, P., Schröter, A., 2013. Teaching students global Software Engineering Skills using Distributed Scrum. In: Proceedings of the 2013 International Conference on Software Engineering. In: ICSE '13, IEEE Press, Piscataway, NJ, USA, pp. 1128–1137.

Paasivaara, M., Vodă, D., Heikkilä, V.T., Vanhanen, J., Lassenius, C., 2018. How does participating in a Capstone Project with Industrial Customers affect Student Attitudes? In: Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training. In: ICSE-SEET '18, ACM, New York, NY, USA, pp. 49–57.

Paiva, S.C., Carvalho, D.B.F., 2018. Software creation workshop: A Capstone Course for Business-oriented Software Engineering Teaching. In: Proceedings of the XXXII Brazilian Symposium on Software Engineering. In: SBES '18, ACM, New York, NY, USA, pp. 280–288.

Palacin-Silva, M., Khakurel, J., Happonen, A., Hynninen, T., Porras, J., 2017. Infusing design thinking into a Software Engineering Capstone Course. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T). pp. 212–221.

Pappas, I.O., Mora, S., Jaccheri, L., Mikalef, P., 2018. Empowering social innovators through collaborative and experiential learning. In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 1080–1088.

Patil, S.P., Neve, J.R., 2018. Productivity improvement of Software Development Process through Scrumban: A practitioner's Approach. In: 2018 International Conference on Advances in Communication and Computing Technology (ICACCT). pp. 314–318.

Penzenstadler, B., Mahaux, M., Heymans, P., 2013. University meets industry: Calling in real stakeholders. In: 2013 26th International Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 1–10.

Pérez-Castillo, R., Caballero, I., Rodríguez, M., 2018. Improving the experience of teaching Scrum. In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 1598–1605.

Pérez-Castillo, R., Caballero, I., Rodríguez, M., 2018. Improving the experience of teaching Scrum. In: 2018 IEEE Global Engineering Education Conference (EDUCON). IEEE, pp. 1598–1605.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic Mapping Studies in Software Engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. In: EASE'08, BCS Learning & Development Ltd., Swindon, UK, pp. 68–77.

Petticrew, M., Roberts, H., 2008. Systematic Reviews in the Social Sciences: A Practical Guide. John Wiley & Sons.

Pirker, J., Kultima, A., Gütl, C., 2016. The value of game Prototyping Projects for Students and Industry. In: Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events. ACM, New York, NY, USA, pp. 54–57.

Portela, C., Vasconcelos, A., Oliveira, S., Souza, M., 2017. The use of industry training strategies in a software engineering course: an experience report. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 29–36.

Potineni, S., Bansal, S.K., Amresh, A., 2013. ScrumTutor: A web-based interactive tutorial for Scrum Software development. In: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 1884–1890.

Prior, J., Connor, A., Leaney, J., 2014. Things coming together: learning experiences in a software studio. In: Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education. pp. 129–134.

Pulko, S.H., Parikh, S., 2003. Teaching 'soft' skills to engineers. Int. J. Electr. Eng. Educ. 40 (4), 243–254.

Reddaiah, B., Reddy, R.P.K., Nagaraju, C., Sree, V.H., 2016. A novel Approach to Adopt Scrum by an Enterprise. In: Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, pp. 645–654.

Rico, D.F., Sayani, H.H., 2009. Use of Agile methods in Software Engineering Education. In: 2009 Agile Conference. pp. 174–179.

Rodriguez, G., Soria, Á., Campo, M., 2015. Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to scrum. Comput. Appl. Eng. Educ. 23 (1), 147–156.

Rodríguez, G., Soria, Á., Campo, M., 2016. Measuring the impact of agile coaching on students' performance. IEEE Trans. Educ. 59 (3), 202–209.

Rodríguez, M.C., Vázquez, M.M., Tslapatas, H., de Carvalho, C.V., Jesmin, T., Heidmann, O., 2018. Introducing lean and agile methodologies into engineering higher education: The cases of Greece, Portugal, Spain and Estonia. In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 720–729.

Rong, G., Zhang, H., Shao, D., 2014. Where does experience matter in software process education? An experience report. In: 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T). IEEE, pp. 129–138.

Root, D., Rosso-Llopart, M., Taran, G., 2008. Proposal based studio projects: how to avoid producing "Cookie Cutter" Software Engineers. In: 2008 21st Conference on Software Engineering Education and Training. IEEE, pp. 145–151.

Rosca, D., 2018. Acquiring professional software engineering skills through studio-based learning. In: 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET). IEEE, pp. 1–6.

Rusu, A., Swenson, M., 2008. An industry-academia team-teaching case study for software engineering capstone courses. In: 2008 38th Annual Frontiers in Education Conference. IEEE, pp. F4C–18.

Santos, A., Sales, A., Fernandes, P., Kroll, J., 2018. Challenge-based learning: A Brazilian case study. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. In: ICSE '18, ACM, New York, NY, USA, pp. 155–156.

Scharff, C., Verma, R., 2010. Scrum to support mobile application development projects in a just-in-time learning context. In: Proceedings of the 2010 Icse Workshop on Cooperative and Human Aspects of Software Engineering. ACM, pp. 25–31.

Schwaber, K., Sutherland, J., 2011. The scrum guide. Scrum Alliance 21, 19.

Scott, E., Rodríguez, G., Soria, Á., Campo, M., 2016. Towards better Scrum learning using learning styles. J. Syst. Softw. 111, 242–253.

Shaw, M., 2000. Software Engineering Education: A roadmap. In: Proceedings of the Conference on the Future of Software Engineering. In: ICSE '00, ACM, New York, NY, USA, pp. 371–380.

Shaw, M., 2003. Writing good software engineering research papers. In: 25th International Conference on Software Engineering, 2003. Proceedings.. IEEE, pp. 726–736.

Sievi-Korte, O., Systä, K., Hjelsvold, R., 2015. Global vs. local - Experiences from a distributed software project course using agile methodologies. In: 2015 IEEE Frontiers in Education Conference (FIE). pp. 1–8.

Sievi-Korte, O., Systä, K., Hjelsvold, R., 2015. Global vs. local -Experiences from a distributed software project course using agile methodologies. In: Frontiers in Education Conference (FIE), 2015 IEEE. IEEE, pp. 1–8.
Smith, T., Cooper, K.M., Longstreet, C.S., 2011. Software Engineering Senior Design Course: Experiences with Agile game development in a Capstone Project. In: Proceedings of the 1st International Workshop on Games and Software Engineering. In: GAS '11, ACM, New York, NY, USA, pp. 9–12.
Smrithi Rekha, V., Adinarayanan, V., 2014. An Open Source approach to enhance industry preparedness of students. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 194–200.
Soundararajan, S., Chigani, A., Arthur, J.D., 2012. Understanding the Tenets of Agile Software Engineering: Lecturing, Exploration and Critical Thinking. In: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education. In: SIGCSE '12, ACM, New York, NY, USA, pp. 313–318.
de Souza, R.T., Zorzo, S.D., da Silva, D.A., 2015. Evaluating capstone project through flexible and collaborative use of Scrum framework. In: 2015 IEEE Frontiers in Education Conference (FIE). pp. 1–7.
Stapel, K., Lübke, D., Knauss, E., 2008. Best practices in Extreme Programming Course design. In: Proceedings of the 30th International Conference on Software Engineering. In: ICSE '08, ACM, New York, NY, USA, pp. 769–776.
Steghöfer, J.-P., 2018. Providing a baseline in software process improvement education with lego scrum simulations. In: 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET). IEEE, pp. 126–135.
Steghöfer, J.-P., Burden, H., Hebig, R., Calikli, G., Feldt, R., Hammouda, I., Horkoff, J., Knauss, E., Liebel, G., 2018. Involving External Stakeholders in Project courses. ACM Trans. Comput. Educ. 18 (2), 8:1–8:32.
Steghöfer, J.-P., Knauss, E., Alégroth, E., Hammouda, I., Burden, H., Ericsson, M., 2016. Teaching Agile: Addressing the conflict between Project Delivery and application of Agile methods. In: Proceedings of the 38th International Conference on Software Engineering Companion. In: ICSE '16, ACM, New York, NY, USA, pp. 303–312.
Stephenson, B., James, M., Brooke, N., Aycock, J., 2016. An industrial partnership Game Development Capstone Course. In: Proceedings of the 17th Annual Conference on Information Technology Education. In: SIGITE '16, ACM, New York, NY, USA, pp. 136–141.
Uskov, V., Krishnaiah, D.B., Kondamudi, R., Singh, U., 2016. Innovative Agile Project Management Curriculum for Engineering Education. In: 2016 IEEE Global Engineering Education Conference (EDUCON). pp. 463–468.
Venson, E., Figueiredo, R., Silva, W., Ribeiro, L.C.M., 2016a. Academy-industry collaboration and the effects of the involvement of undergraduate students in real world activities. In: 2016 IEEE Frontiers in Education Conference (FIE). pp. 1–8.
Venson, E., Figueiredo, R., Silva, W., Ribeiro, L.C., 2016b. Academy-industry collaboration and the effects of the involvement of undergraduate students in real world activities. In: 2016 IEEE Frontiers in Education Conference (FIE). IEEE, pp. 1–8.
Villavicencio, M., Narvaez, E., Izquierdo, E., Pincay, J., 2017. Learning scrum by doing real-life projects. In: 2017 IEEE Global Engineering Education Conference (EDUCON). IEEE, pp. 1450–1456.
Vu, J.H., Frojd, N., Shenkel-Therolf, C., Janzen, D.S., 2009. Evaluating test-driven development in an industry-sponsored capstone project. In: 2009 Sixth International Conference on Information Technology: New Generations. pp. 229–234.
Wallace, C., Mohan, S., Troy, D., Hoffman, M.E., 2012. Scrum Across the CS/SE Curricula: A retrospective. In: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education. In: SIGCSE '12, ACM, New York, NY, USA, pp. 5–6.
Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requir. Eng. 11 (1), 102–107.
Williams, J.J., Kim, J., Keegan, B., 2015. Supporting instructors in collaborating with researchers using MOOClets. In: Proceedings of the Second (2015) ACM Conference on Learning @ Scale. ACM, pp. 413–416.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. Springer Science & Business Media.
Wong, K., 2016. Experiences in Constructing a MOOC Specialization. In: Proceedings of the 21st Western Canadian Conference on Computing Education. In: WCCCE '16, ACM, New York, NY, USA, pp. 19:1–19:4.
Xiao, D., Miller, R.C., 2014. A multiplayer online game for teaching software engineering practices. In: Proceedings of the First ACM Conference on Learning@ Scale Conference. ACM, pp. 159–160.
Zazworka, N., Stapel, K., Knauss, E., Shull, F., Basili, V.R., Schneider, K., 2010. Are developers complying with the process: an XP study. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, p. 14.
Zhang, H., Babar, M.A., Tell, P., 2011. Identifying relevant studies in software engineering. Inf. Softw. Technol. 53 (6), 625–637.
Zhang, X., Hu, T., Dai, H., Li, X., 2010. Software development methodologies, trends and implications: A testing centric view. Inf. Technol. J. 9 (8), 1747–1753.

Orges Cico holds a Bachelor and MSc in Computer Engineering from Politecnico di Torino, Italy, and is currently a Ph.D. Candidate at the Norwegian University of Science and Technology. His research topic is in Empirical Software Engineering, more precisely in Lean Software Startups and Software Engineering Education. He has published over ten peer-reviewed papers in high-quality international conferences such as ICSE, EASE, EDUCON and FIE. Previously he was a Full-Time Lecturer at the Canadian Institute of Technology, Tirana, Albania, and the Director of the Metropolitan Incubator Center (NGO part of the University Metropolitan Tirana, Albania). In the last eight years, he has had extensive experience in both industry and academic settings. He has participated in several project development dedicated to cloud, mobile, and IoT systems and taught subjects related to Advanced Software Engineering and Operating System Design.

Letizia Jaccheri (Ph.D. from Politecnico di Torino, Italy) is Professor at the Department of Computer Science of the Norwegian University of Science and Technology. Jaccheri's research is on: software engineering; entertainment computing; computational creativity; ICT-enabled social innovation. Jaccheri is the Norwegian representative and Vice President of IFIP TC14 on Entertainment Computing. She has published more than 200 papers in International conferences and journals. She has been teaching courses in software engineering at various levels since 1994. She has supervised PhD students, Post-doctoral students and acted as opponent for national and international defences. From 2015 to April 2018 she was independent director of Reply S.p.A, an IT company with 6000 employees worldwide. She has been general chair of IFIP ICEC 2015, co-chair of ACM IDC 2018, and Program Chair of the European Computer Science Summit 2018. She participates to several Horizon 2020 projects, among which INITIATE INnovation through big data and socIal entrepreneurship; UMI-Sci-Ed Exploiting Ubiquitous Computing, Mobile Computing and the Internet of Things to promote STEM Education; SOCRATIC SOcial CReATive IntelligenCe Platform for achieving Global Sustainability Goals.

Dr. Anh Nguyen-Duc, an Associate Professor at University of South Eastern Norway, Bøcampus. He has accumulated his research experience in the area of empirical software engineering, particularly cybersecurity and software-intensive business. In the last five years his research focus is on Minimum Viable Product and software startup. He has more than 60 peer-reviewed publications in SE journals (JSS, IST, ESE) and SE conferences (ICSE, ESEM, XP, EASE). He has three edited books in business-driven software engineering. He has served as a chair (workshop/ conference chairs) in 10 organization committees, 28 program committees as well as reviewers, and guest editors on recognized journals.

He (Jason) Zhang is a Full Professor of Software Engineering and the Director of DevOps+ Research Laboratory at the Nanjing University, China, also a Principal Scientist with CSIRO, Australia. He was awarded B.Eng (NUAA), M.PM (USyd) and Ph.D (UNSW). Prof. Zhang is an internationally recognized researcher and expert in software engineering. He joined academia after seven years working in software industry, developing software systems in the areas of aerospace and complex data management. He undertakes research in software engineering, in particular software process (modeling, simulation, analytics and improvement), software architecture, software security, blockchain-oriented software engineering, empirical and evidence-based software engineering, service-oriented computing, and data-driven software engineering. He has published over 150 peer-reviewed papers in high quality international conferences and journals, and won 11 Best/Distinguished Paper Awards from several prestigious international conferences and journals. He is a Member of the IEEE Computer Society and the ACM (SIGSOFT), a Senior Member of China Computer Federation (CCF), and serves on the Steering Committees and Program Committees of a number of high quality international conferences in software engineering community.

Please cite this article as: O. Cico, L. Jaccheri, A. Nguyen-Duc et al., Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. The Journal of Systems & Software (2020) 110736, https://doi.org/10.1016/j.jss.2020.110736.