How industry trends influence software engineering education: a mapping study

Abstract—**Contemporary approaches and trends in software engineering courses have been continuously updated over the last four decades. Adaptation to industry needs is crucial for future educational purposes and vice versa. Tech startups have become a driving force in the economy and a major industry trend. The goal of this paper is first to critically assess how contemporary industry trends and in particular, tech startups have influenced the updating of software engineering curricula. The second goal is to evaluate the contribution of industry and in particular, tech startups, processes and models in present learning approaches. The third goal addresses how stakeholders have helped in the infusion of industry trends and in particular, in tech startup approaches in academia. This study is a systematic literature mapping. A total of 138 papers were selected based on education goals, research, and contribution type. Of the primary education topics, 78% were related to teaching strategies, 9% to globalization and training methods, and less than 5% to tech startup and industry innovation. Common stakeholders accounting for the change are students, researchers, and lecturers and project managers, product owners, customers, and clients from industry. This study showed that industrial models or methods involving Agile and Scrum have been widely adopted. Less investigated areas that have recently become common industry trends, such as tech startup models adopting lean methodology, require further attention and might create opportunities for updating the curricula. We discuss future model possibilities for exploiting tech startups as a means for renewing future capstone courses.**

Index Terms—**software engineering courses, teaching approaches, mapping study, tech startups**

I. Introduction

Software engineering (SE) learning approaches have continuously evolved over the last few decades [1]. The driving force has primarily been the industry demand for skilled employees [2] and academic concern about teaching relevant topics [3, 4]. Important recommendations have long been available from the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) [5], including extensive experience in software engineering curricula. Efforts are also being made in the context of Computing Curricula of 2020, where SE plays an important role [6]. However, emulation of the industry environment within the academic setting still poses challenges to overcome. The following gaps were identified in [7]: lack of a real product, short duration, high turnover, low sophistication, no or little maintenance, and missing customers. Academia has tried to fill these gaps, as well as other issues, related to student readiness to fulfill industry requirements, mentioned in [8], by developing courses that have a longer duration or rely on capstone projects or flipped classrooms. In these cases, graduate student projects in software engineeringrelated courses adopt the idea of prototyping through industry customer-driven [9, 10], free and open source software (FOSS) [11], play money [12], startup, innovation, and creativity-driven [13], on-site or online training platforms [14, 15] or other models or methods exploiting agile or scrum methodology at the local or global scale [16, 17]. However, the non-realistic toy project has been considered to bring more harm [18] than help.

The software industry is also a driving force of the economy. Not all industry approaches, however, have the same performance. A recent trend in the industry is tech startups, with high growth potential [19], and supported by the lean startup method [20]. Tech startups adopt rapid prototyping by developing a minimum viable product (MVP), which is then fine-tuned according to business needs. Lean methodology as a business practice has also been proven to be effective in software engineering classes [21].

Recent publications [22, 23, 24] suggest that software engineering students should be combined with entrepreneurship students into interdisciplinary teams that act on a tech startup-driven model. This approach, despite its challenges (intellectual property, legal, etc.), might be an emerging candidate for motivating students to deliver more realistic products. It promotes the recommendations of the Agile Manifesto [25], for face-to-face communication and external pressure. Moreover, the set of skills for software developers has drastically changed over time. It requires understanding of not only coding in a polyglot pattern but also teamwork and creative and innovative thinking.

The primary goals of this paper are to identify the influence of contemporary industry trends and in particular tech startups, as an economic driving force [64] in software engineering in education (SEE), identify the inclination of the SEE toward models or processes adopted from industry and in particular, tech startups, identify how the primary stakeholders have influenced the adoption of the SEE curricula for industry and in particular, tech startup requirements, and identify future tech startuporiented models, frameworks, and relevant stakeholders, inducing further updating SEE curricula, in addition to narrowing the gap with industry requirements.

The 138 papers selected for review were classified by the SEE goal, research, and contribution type. Publication trends and sources were also evaluated to understand the evolution and quality of the research conducted.

We examined the efforts made and research gaps in tech and software startups inclusion in academia [26, 24, 27, 28] and in the adoption of lean methodology as a teaching method. This helped us pinpoint an encouraging future research direction, apart from evaluating a broad overview of the state-of-the-art trends in SEE influenced by industry.

In the second section, we explain the systematic mapping approach we used to select the papers. In the third section, we analyze the results and answer the research questions. In the fourth section, we discuss the findings and the limitations of the research. In the fifth section, we conclude and propose future research opportunities.

II. Background

The software engineering discipline has evolved during the past 40 years, guided by the Software Engineering Body of Knowledge (SWEBOK) [29]. Academia has constantly funneled computer science students to the industry with appropriate skills to excel in their jobs.

However, the software industry has transitioned to adapting to market needs, in providing high-quality products within a reasonable time to market. Agile methodology has proved efficient and has overcome some important challenges (e.g., face-to-face communication and requirement iteration).

During the last two to three decades, a trend in the software industry have been tech and software startups. In most cases, lean methodology leads to early exploration of the market needs and development of the MVP. The two are discussed further in the next sections.

1) Startups: A startup is commonly defined as an entrepreneurial or business venture. The primary goal is to meet a marketplace need by developing a viable business model for products, services, processes, or platforms. The failure rate of startups is commonly high; however, successful startups have had a major impact on the industry [30]. Startups undergo several development phases: Ideation (Product or Service idea), Concepting (Mission and Vision), Commitment (Team with the initial product), Validation (Iteration and testing the initial idea), Scaling (Focus on key performance indicators), and Establishment (Increasing growth and market potential).

The ecosystem is a primary factor in startup survival and growth. It usually consists of different entities such as incubators, entrepreneurial schools, accelerators, large companies, government programs (embodying mentors or advisors), and local or global markets. The ecosystem usually fits the startup development stage. Funding is also crucial for the survival of a startup. Funding is commonly based on self-contributions, in the form of self-investment (by bootstrapping between jobs) or loans (from relatives or friends). Other funding options in the early stage of startup formation can come from pre-seed or crowd funding. In later stages, when an MVP has been developed and iteration with the market is a must (do or die approach), the need for larger funding amounts from venture capitalists (VCs) and angel investors (AIs) becomes obvious. Finally, if the startup has developed a fully operational product or service, then the market, either local or global, decides the startups growth potential.

Figure 1 gives a broad overview of the startup development phases, funding options, and survival ecosystem.

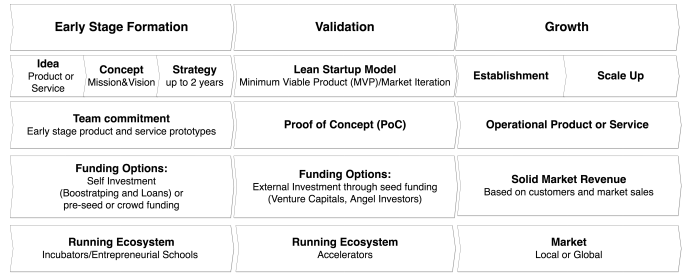

Fig. 1. Startup ecosystem, development and lifecycle phases

2) Lean Startups: A lean startup is a methodology for developing businesses and products. It focuses on shortening the product development cycle, through iterative product releases, market experimentation, and validation. Meeting the needs of early customers should reduce later failure risks of large investments. Teams that adopt a lean startup strategy usually develop a continuously changing MVP [31].

3) Student context: Software engineering students usually focus primarily on computer science. Their careers are oriented toward well-established and large companies, where training is common for providing software engineering skills (Languages and frameworks, Tools, Program understanding, Programming experience) for lower-level employees. Decision making is usually part of senior developers or project managers work scope. This leaves little space for them to invest in soft skills (resume building, career planning, communication, teamwork and collaboration, time management, presentation planning, and dealing with learning challenges). Participating or embarking on startup methodology, however, helps students become problem solvers, as well as make fast leadership decisions, through face-to-face communication with end customers. Thus, these employees better develop their soft and technical skills in parallel.

4) Research context: In this paper, we analyze how contemporary industry trends and specifically, tech startup models have influenced the updating of teaching approaches providing a valuable contribution to the industry. Finally, we analyze how commonly accepted stakeholders have participated in improving the process. Figure 2 shows the research perspective we adopt in this mapping study.

An examination and an elaboration of approximately 100 text snippets extracted from about 20 initially reviewed papers are used to obtain major categories inherent in the research context and questions, which are utilized in section three.

III. Mapping Study

Petersen et al. [32], suggested a systematic mapping study provides a structure for the type of research reports and results that have been published by categorizing them. The first step of the process involves posing research questions, which then help generate a visual summary of the research results. The other steps involve screening based on keywords, abstracts, metadata, etc. The results help answer the research questions. The primary focus of a systematic mapping study

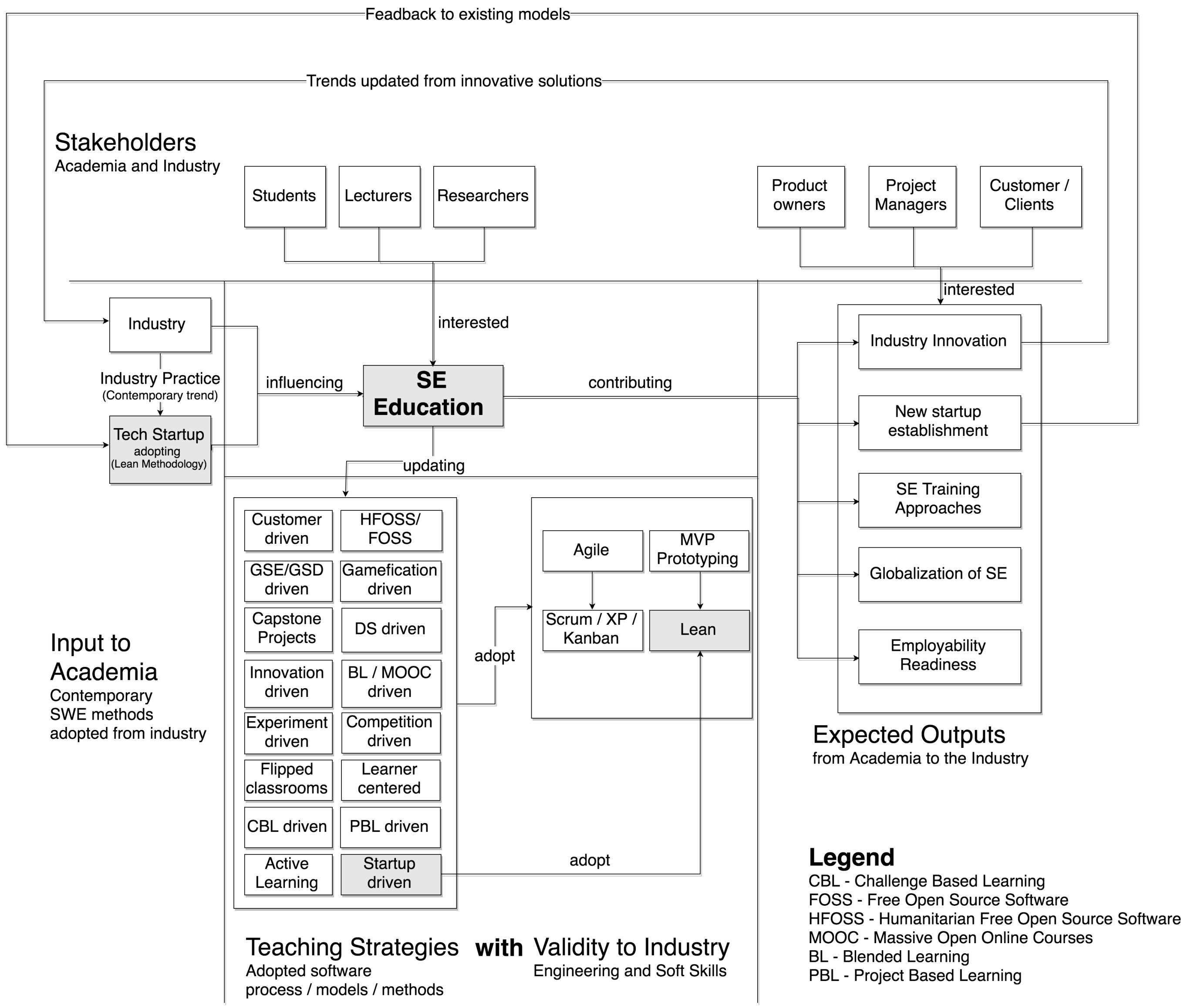

Fig. 2. Research perspective utilized in conducting the mapping study

is to identify gaps in the research area under investigation. Figure 3 represents the systematic mapping process [32].

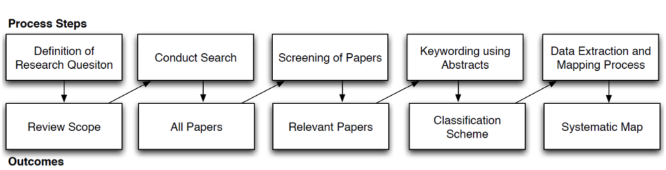

Fig. 3. The Systematic Mapping Process [32]

Systematic literature mapping (SLM) relies on five process steps. The initial step is to define the research scope. The next two steps mainly involve searching and filtering results based on a strict set of screening criteria. The last two demand a proper analysis and mapping of the result so that the research questions are answered.

A. *Research Questions and Search Strategy*

The first goal of this study is to analyze publications concerning SEE to show the present approaches and best learning practices adopted within the last 5 years. The focus is how industry trends (in particular, tech startups) have influenced learning methods for improving student skill readiness for the industry. We defined six research questions that are reported in Table I.

The primary sources used in this study are reported in Table IX. The sources are those most commonly used in investigating software engineeringrelated publications.

According to the systematic mapping process, conducting research involves identifying the search string. Population, intervention, comparison, and outcome (PICO) criteria are

TABLE I
RESEARCH QUESTION

RQ#	Research Question	Motivation
RQ1	How have contemporary industry trends and in particular tech startups influenced teaching strategies?	This questions tries to identify what has been the impact of one the growing trends in industry [19] on software engineering education. We have limited the research to mainly articles published in the past five years so that only state of the art approaches derived from cutting edge industry models are considered.
RQ2	How processes/methods adopted from industry and in particular tech startups, adopting lean methodology, have influenced software engineering courses?	This questions tries to identify which processes and methods adopted from software engineering curricula have been utilized the most, and identify the lean startup model influence.
RQ3	How have stakeholders influenced in enhancing software engineering curricula according to industry best practices?	The question tries to identify if all important stakeholders have been considered when enhancing the curricula.
RQ4	What are the recently proposed research and contribution type?	The fourth research question provides information about the proposed solutions.
RQ5	How publications have evolved over time? What are the research and publication trends?	The fifth question shows the evolution of the publications concerning the subject under study.
RQ6	In which bibliographical sources are they published?	The sixth question evidentiates the different sources where articles concerning learning methods have been published. This consideration is important to evaluate the actual impact the research has had in SEE.

TABLE II
MAIN SOURCES UTILIZED FOR THE MAPPING STUDY

Source Type	Denomination
Digital Libraries	IEEE-Xplore, ACM Digital Library
Databases	Science@Direct, Wiley InterScience

good options, according to Petersen et al. [32] driven by the research questions. The search string should contribute to maximum article coverage, within a manageable size. RQ1 and RQ2 suggest a broad overview of the research area. This is an important consideration for the mapping study, considering most recent teaching practices and adopted models. Moreover, an overview would help to better understand how much software industry trends (tech startups adopting lean methodology) presently influence academia. RQ3 identifies the stakeholders that commonly influence change. To build a composite search string to evaluate the stated research questions, Boolean operators were used, resulting in the following outcome:

(Software Engineering Course OR Software Engineering Education OR Software Engineering Training) = (SUBS1) AND (agile OR mvp OR prototyping) = (SUBS2) AND Scrum OR XP OR Kanban OR Lean = (SUBS3) AND (customer driven OR tech startup driven OR lean startup driven OR innovation driven OR creativity driven OR global software engineering driven OR gamification driven OR training driven OR free open source software driven OR flipped classroom OR experiment driven OR MOOC driven OR competition dr OR capstone project) = (SUBS4) AND (students OR researchers OR lecturers OR teachers OR project managers OR product owners) = (SUB5).

The search string was constructed based on correspondence to the research questions in Table I so that proper evidence could be collected. The mapping between the substrings (SUBS - keyword groups) of the search string and the research questions is reported in Table III.

TABLE III
RESEARCH QUESTION VS. SEARCH STRING KEYWORDS

RQ#	SUBS1	SUBS2	SUBS3	SUBS4	SUBS5
RQ1	x			x	
RQ2		x	x		
RQ3					x
RQ4	x	x	x	x	x
RQ5	x	x	x	x	x
RQ6	x	x	x	x	x

The search process took place from May to August 2018. The same search string was applied to the same meta-data (title or complete text) of each paper from all the sources reported in Table IX.

Publications were restricted those published within the last 5 years (2014 to 2018). This timeframe is discussed in the section on the limitations of the study and their threats to validity trade-off.

B. Screening papers for inclusion and exclusion

The screening process identified the most relevant articles based on the research questions in Table I 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 inclusion and exclusion criteria are reported in Table IV.

A total of 1363 articles published during the past 5 years were identified. Most of the publications were found in the ACM/IEEE digital libraries. Phase 1 involved further filtering based on the inclusion and exclusion criteria (remaining 163 articles). To ensure we did not overlook any important materials, additional searches, guided by the established criteria, were performed directly on key conference and workshop proceedings (ICSE, ITiCSE, ICGSE, and SEEM) and others reported in Table IX, producing a further 14 studies. In phase 2, duplicate articles (16) were removed, and another 23 articles based on the abstract, subtitles, or full paper content were found in phase 3. The remaining articles totaled 138, which can be retrieved from [33]. All the steps and statistics are shown in Figure 4. Further the mapping involved assignment of keywords based on the abstract whenever it was clearly stated and in particular cases the review involved also reading of the introduction and conclusion sections.

TABLE IV
SCREENING CRITERIAS FOR THE PAPERS

Included (when meeting 1 out of 4 criterias)	**Excluded (when meeting 1 out of the 3 criterias)**
Studies related to SEE, industry training or courses adopting one or more of the teaching strategies (customer-driven, innovation driven, globalization driven etc.).	Studies not concerning software engineering but discussing the techniques in a different engineering context.
Studies related to SEE, industry training or courses adopting one or more of the software processes/models or methods.	Studies that tangentially mention software engineering for different scopes not directly related to education.
Best practices and models in Software Engineering reported from conference and open source journal publications that impact education enhancement.	Studies concerning SEE not particularly applied to education in combination with industry.
Studies that include industrial training or innovation and tech startup as part of the teaching process.	N/A

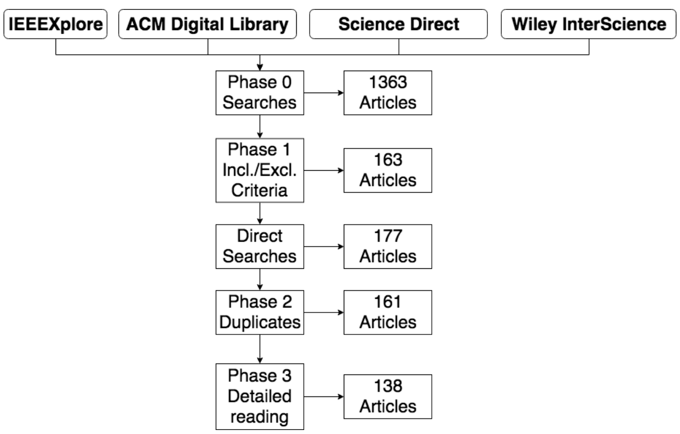

Fig. 4. Statistics of the selections based on three filtering phases [32]

C. *Classification Scheme*

To classify the articles, we used the scheme proposed by Petersen et al. [32], who rated classification schemes based on a set of quality attributes [32]: Orthogonality (there are clear boundaries between categories defined based on existing literature), Based on the terminology used in literature (complete. No categories are missing), and Accepted (the community accepts and knows the classification/taxonomy).

During the process, we tried to use keywords that were very close and relevant to the research questions. In Figure 5, we illustrate the classification facets. The first (SEE Goal) is a derivation based on the existing literature in the research context Figure 2. Tech startups adoption of lean methodology has played a key role in classifying the expected outputs from SEE. The influence within the academic context is part of the teaching methods, also influenced by the industry. The other facets are closely related to classification of the primary stakeholders and models, processes, or methods inherent in industry modern or contemporary trends, including tech startups. The last two were also proposed in [32].

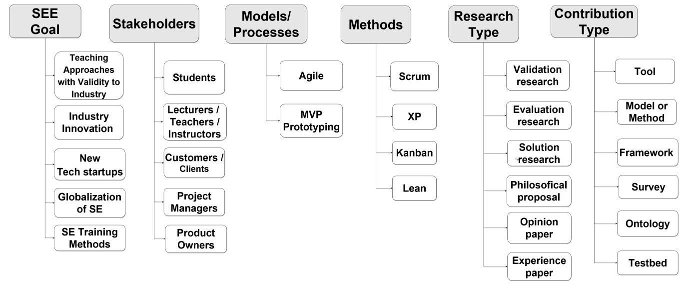

Fig. 5. Classification Scheme

1) SEE Goal: There are different approaches for classifying the education goal in SE courses. SEE is often guided either by industry patterns, models, methods, and aims of enhancing student comprehension, as well as improving students soft and engineering skills demanded by the industry. For this study, we focused on particular aspects of the SEE goal facet related to their validity in the industry by filling the gap in academia, reported in Table V.

TABLE V
SEE CATEGORIES

Category	**Properties**
Teaching strategies with validity to industry	Improved comprehension of industry methods such as agile (scrum), prototyping, lean and teaching strategies capstone projects, gamification, customer-driven, free open source software, inverted classrooms etc. Soft, programming, engineering skills embracing validity to industry needs.
New Tech Startups	Entrepreneurship, High growth, Success Stories, Prototyping, Lean Methods.
Globalization of SE	Multinational vs. local environment.
Industry Innovation	Ideas, Creativity, Product, Services, Business Models, Methods.
SE Training Methods	Developing skills with focused tech oriented online Resources eg. MooC etc. Academic courses incorporated with company training classes.

2) Stakeholders, Models or Processes, and Methods: The Stakeholders, Model/Processes, and Methods facet is constructed taking into consideration modern industry and primarily tech startup approaches. They represent a subset of the existing models or processes (Agile, waterfall, iterative, spiral, etc.) or methods. The same applies to the stakeholders that are common in the literature.

3) Research type: The research type facet was based on the schema proposed in [34]. It usually contained six categories.

- *Validation research:* Validation research is not implemented in practice and focuses on the validation of the solution in the lab or simulation scenarios.
- *Evaluation research:* Evaluation research is implemented in practice and shows the solution implementation and what the consequences of the implementation in an environment are in terms of benefits and drawbacks.
- *Solution proposal:* A solution proposal is a new technique or a significant extension of an existing technique.

- *Philosophical proposal:* This proposal shows a point of view regarding the subject without the preciseness of a solution proposal.
- *Opinion paper:* An opinion paper reports the authors opinion of what is good or bad.
- *Experience paper:* An experience paper reports on personal experiences from a real-life project.

4) Contribution type: The contribution type categories were derived from Petersen et al. [32]. We adopted the following for this mapping study:

- *Tool:* Papers proposing a tool related to SEE.
- Model or Method: Papers describing a new or existing model or method. Agile and Prototyping are identified as models, while Scrum, XP, and Kanban are identified as methods in SE.
- *Framework:* Papers proposing frameworks in SEE.
- *Survey:* Papers exposing techniques utilized in SEE but in which no solution is proposed.
- *Ontology:* Papers proposing an ontology for identifying and discussing the information that will be exchanged to preserve SEE.
- *Testbed:* Papers proposing a testbed that enables researchers to study different aspects of SEE.

IV. Analyzing Results for answering RQs

We classified the papers based on the categories described in each facet to discuss the six research questions. The findings related to the research questions are discussed in the following subsections.

RQ1: How have contemporary industry trends and in particular tech startups influenced teaching strategies?

To answer this question, the SEE goal facet, discussed in the Background section, was created. The results are shown in Table VI. Most of the papers deal with teaching strategies that are valid for industry soft or technical skills (77.5%). The authors discussed different models or methods for improving SE students understandability of industry practices. Some of the commonly used techniques were problem/project driven learning (PBL) [35] and customer-driven based on FOSS [36] adopted to Capstone project courses [37, 38], play/gamification of course content [39, 12], and flipped classrooms [40].

TABLE VI
Number of papers by education goal in SEE

SEE Goal	Number of papers	Percentage
Teaching strategies with Validity to the Industry	106	77.5%
New Tech Startups	6	4.5%
Industry Innovation	1	0.8%
SE Training Methods	12	8.6%
Globalization of SE	12	8.6%

Validity for industry is a major indicator when teaching strategies are proposed. Most publications discussed several aspects related to connecting classroom and industry, for technical and soft skills.

Training as an additional asset of SE education was also discussed in [41]. Training approaches have been considered to have important value when they are conducted at company sites [15, 42] within the education setting. A common approach is online training through massive open online course (MOOC) [14].

Globalization of SE, involving global software engineering (GSE) or global software development (GSD), is one of the most recent trends encountered in the literature. Beecham et al. reported an eminent need to address this teaching strategy in recent publications [43, 44]. Others suggested how Agile through Scrum methodology can adopt to GSE/GSD [45].

Tech startups as part of education for millennials [46] have been indicated to be a new emerging strategy primarily emphasized by Buffardi et al. Publications [23] emphasized the realistic education setting obtained with this approach. The tech startup model has been experimented with by software engineering and entrepreneurship students in [47], where further contribution to new tech startup formation has been claimed. Industry innovation in the context of SEE has barely made any realistic contribution.

During the investigation process, we identified the major approaches used in SEE based on best practices in industry trends, answering RQ1.

RQ2: How models/processes adopted from industry and in particular tech startups, adopting lean methodology, have influenced software engineering courses?

The most recently adopted model in industry and academia is Agile, commonly combined with Scrum methodology. This was reported in more than 70% of the reviewed publications. The context in academia is multifold, including GSE [17, 48], inverted classrooms [49], capstone projects [50] and even SEE for millennial students including tech startups [51]. Other methods, such as Kanban or XP, have been adopted in capstone courses [50].

Lean methodology adopted from business is becoming a new trend for improving SEE [28] usually combined with tech startups and emphasized by Heggen et al. for millennial students [46].

During the investigation, we observed that most recent publications in 2017 and 2018 addressed tech startups as a new important trend for updating SEE curricula, answering RQ2.

RQ3: How have stakeholders influenced in enhancing software engineering curricula according to industry best practices?

To answer RQ3, we had to review most of the publications. The primary actors were identified from flipped classroom models, capstone courses relying on project-based learning, global SE or customer-driven courses involving mainly students, (academic or industry) instructors, teachers, or coaches, project managers/product owners, industry professionals/project leaders, and end customers or clients. When an industry role is missing, it is either covered by the internal university staff (e.g., clients) [52] or the course instructors. This simulation was partially efficient due to the lack of real external pressure. However, a better proposal was made by Heggen et al. in [46], where the course blends internship and summer jobs with the SE course, in order to provide millenial students with useful soft and technical skills. Moreover, it is important to understand that real projects have many constraints and involve more actors, and simulation with pseudo-budgets or play money [12] or prototyping [53] might not be enough. Software end product quality and maintenance are important aspects of SE. Few considered the impact of the realistic aspect of the course setting in delivering appropriate teaching strategies. Moreover, the proposed models should take care that simulations provide realistic stakeholder participation to obtain valid learning enhancement. Tech startups, moderately investigated for millennial students, as reported by Buffardi et al. in [22, 23, 51], have a strong foundation in adopting lean and Scrum methodologies. They are a promising industry trend, and other stakeholders (entrepreneurs, innovation centers, and accelerators) could become part of SE courses. A good mapping of the different actors was provided in [54] where external industry stakeholders are considered to play a key role.

RQ4: What are the recently proposed research and contribution type?

Some of the results from the contribution and research type facet are presented in Table VII. When considering the classification based on contribution type, models or methods were the most common scenario encountered in the publications (72.3%), and they were primarily matched with validation research. This means that most models or methods were evaluated within the studies by simulating industry scenarios with an experimental classroom setting. Holmes et al. [35] proposed experimenting with free open source software. Bruegge et al. [10] emphasized the importance of conducting customer-driven courses with real industrial clients. Other authors experimented with gamification [55], global SE [45], innovation [24], as well as tech startups adoption of lean methodology [23, 56].

Models or methods have been implemented in real-life case studies with positive outcomes. For example, [11, 57] proposed FOSS in conjunction with industry projects to enhance students soft and technical skills. [58] proposed the use of a maturity model adopted from industry within a software engineering course. Many similar proposals showed a shift in the models adopted, long utilized in industry within the SEE curricula. The other papers involved surveys conducted to derive theories from empirical data. Common use of such theories was then encapsulated within the teaching strategies reported in the mapping done later in the section. For example, [50] analyzed the outcome of capstone projects for students. Around 6% of the papers were related to tools, involving serious gaming [59], course assessment [60], and global software engineering [61]. However, almost no publications were related to testbeds or ontologies. These papers represented means, exchange of knowledge, or establishment of ecosystem to develop further or preserve/transfer SEE knowledge.

TABLE VII
NUMBER OF PAPERS BASED ON CONTRIBUTION TYPE.

Contribution type	**Number of papers**	**Percentage**
Model or Method	100	72.3%
Framework	12	9%
Survey	17	12%
Tool	8	6%
Ontology	0	0%
Testbed	1	0.7%

Classification based on research type indicates that many publications were related to validation research (62%), followed by evaluation research and solution proposals, shown in Table VIII. Prototyping solutions have been proposed, and scholars attempted to validate the prototypes within isolated classroom environments. For example, [40] validated Scrumage within a class experiment. Others simulated scenarios in the global software engineering setting [62, 63] relied on bare experimental constraints and exploited different approaches (gaming or scrum) for capstone projects. However, many case studies were also conducted in correlation with industry demands and actual implementations evaluated in real-life scenarios. They mainly represented evaluation research (12%) conducted in strong collaboration with industry. For example, [64] emphasized with a real case study project academyindustry 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 academia in a progressive manner. Other publications implemented collaborations through industry instructors [52] or reported experiences in considering internal university entities as customers [65], in capstone courses.

Philosophical and opinion papers made up the smallest number of publications.

TABLE VIII
NUMBER OF PAPERS BASED ON RESEARCH TYPE.

Research type	**Number of papers**	**Percentage**
Validation research	86	62%
Evaluation research	17	12%
Solution Proposal	12	9%
Philosophical paper	0	0%
Opinion Paper	3	2%
Experience paper	20	15%

To observe the relations between the different facets, we mapped the results for one facet against those for another and present the outcomes in bubble plots reported in Figure 6. The SEE goal is compared against the contribution and research proposals. Most papers described teaching strategies that were valid for the industry through models or methods combined with validation research. Although the tech startup model relying on lean methodology is a recent trend, very little research has addressed this trend.

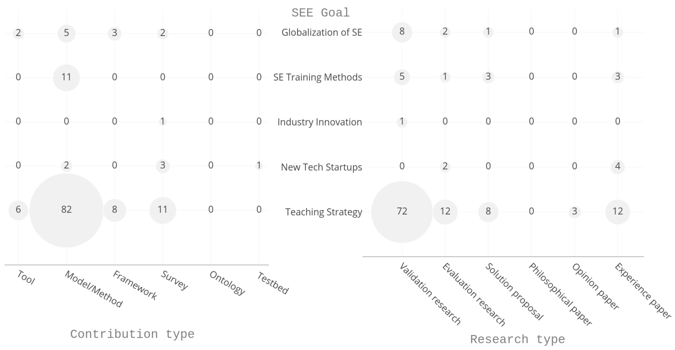

Fig. 6. Number of papers by research, education type and SEE goal facets

RQ5: How publications have evolved over time? What are the research and publication trends?

Based on our search of the last 5 years, from 2014 to 2018, about 50% (65) of the papers were published in 2017 and 2018, Figure 7.

The number of publications declined slightly in 2015 but then increased quickly in 2017 and 2018. This trend was observed in almost every single category: validation research, evaluation research, solution proposals and philosophical, opinion, and experience papers.

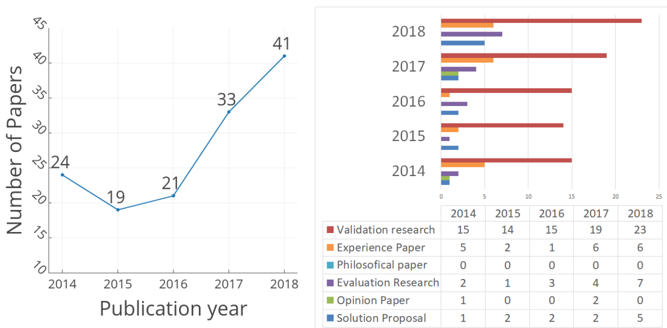

Fig. 7. Number of papers by year and research type.

RQ6: In which bibliographical sources are they published?

The main sources were conferences and workshops, about 85%, and the remaining, about 15%, were research journals. There was a wide range of sources for the publications, but high-quality software engineering and computer science education conferences or journals provided most relevant papers for this study. Publications in journals such as ACM Transactions on Computing Education (TOCE) and Journal of Systems and Software have discussed topics important topics in the involvement of industrial clients, as well as efficiently implement agile methods and teamwork in SE courses [10]. Other publications in IEEE Software and British Journal of Education Technology put focus on Global Software Engineering as an imminent need to prepare future SE students [44] and MOOCs as a tool that can transform SE learning on a global scale [14].

Tech startups [22] and adopting lean methodology in SE courses [28] 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 International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET) in 2017.

Table IX provides an overview where part of the 138 selected papers were published. We checked the quality of the most important sources based on the Journal Citation Reports (JCR) [66], indicating the journal impact factor (JIF), the Norwegian register of scientific journals, series and publishers (NSD) [67], indicating journal and conference scientific level (SL), and the Scimago Journal Rank [68], indicating the journal or conference H-index.

The quality reports validity is considered between the 2014-2017. Due to the large number of conferences we advise the reader to consult the full list from [33].

TABLE IX
SELECTED PAPERS SOURCES

Journal/Conference/Workshop	**Papers Number**	**JIF [66]**	**SL [67]**	**H-index [68]**
Educational Technology Research and Development	1	1.728	1	75
IEEE Transactions on Education	2	1.600	1	57
IEEE Software	2	2.879	2	96
Journal of Engineering Education	1	1.976	1	86
Journal of Systems and Software	3	2.278	2	89
ACM Transactions on Computing Education (TOCE)	3	1.535	1	22
British Journal of Educational Technology	1	2.729	1	76
Computers in Human Behavior	2	3.536	1	123
International Conference on Software Engineering Education and Training (CSEE&T)	8	-	1	6
Frontiers in Education Conference (FIE)	9	-	1	32
International Conference on Global Software Engineering Workshops (ICGSEW)	3	-	-	1
Global Engineering Education Conference (EDUCON)	6	-	1	14
International Workshop on Software Engineering Education for Millennials	3	-	-	-
SIGCSE Technical Symposium on Computer Science Education	2	-	-	22
International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)	6	-	1	39
International Conference on Software Engineering	3	-	1	118

V. DISCUSSION

By conducting Systematic Literature Mapping , we identified important fields that are commonly addressed by most of the researchers, including gamification, learning by doing, and real customer projects as part of SE courses to boost students comprehension of the courses.

The main finding is the adoption of industry-relevant methods as part of the teaching approaches (e.g., Agile, Scrum, XP, etc.) was common.

Validation and evaluation studies appeared to be more common and involved case studies or implementations. Experience and solution proposals made up a significant part of the contributions.

Innovation and startup driven publications make up a smaller chunk of interest, and this might be particularly critical for SEE following industry trends [19] as the validity to industry of the courses should be continuously addressed. Similar mapping studies [69] showed the present startup diffusion and challenges in industry.

A. Limitations of the Study and Threats to Validity Trade-off

This study suffers from several threats to validity. In terms of time and data validity, the dataset was collected from state-of-the-art publications only.

Conclusion validity occurred due to subjective criteria defined by the researcher. The research and conclusions were based on expectations. The construct validity mainly related to identifying primary papers only, or a lack of further investigation in other digital resources. Internal validity due to data analysis based on the systematic mapping of abstracts can be mitigated from a more complete literature review. External validity can occur when generalizing results. We drew conclusions based on the mapping study only, so this threat was mitigated [70].

VI. Conclusions and future work

In the paper, we performed a systematic mapping study of the literature in software engineering in education primarily based on tech startups. The research questions helped us observe that common trends, such as customer driven, gamification driven, learning by doing, innovation driven, and problem/project-based learning, have largely been adopted in combination with capstone, flipped classroom, or blended learning courses.

A thorough analysis was conducted by reading the different papers and classifying them based on the SEE goal, research type, and contribution type facets. In most cases, we found that teaching strategies for students throughout the models, methods, and surveys were the primary goal. However, emerging trends in the software industry, such as tech startups adopting lean methodology, are becoming part of SEE. Recent conference publications have posed important unexplored gaps for future research, such as, how to converge tech startup utilizing lean methodology in SEE? What challenges will students encounter, and how to tackle them? How have the skillset and employability requirements changed for millennial students?

A focus in this area might lead to important research, where students can actually develop appropriate skills, due to the many other realistic stakeholders involved in this model. Entrepreneurs, Venture Capitalists (VCs), Angel Investors (AIs), and many other parts of incubation and acceleration centers should be mentioned. All put pressure to create successful prototypes with limited time and resources.

While observing continuous integration between teaching strategies and existing models or methods, we propose a combined model among GSE and tech startups. For successful startups it is inevitable that during some point in the lifecycle to enter a high-growth phase. The model would raise the question: How can tech startups foster Global Software Engineering as a tool for teaching realistic capstone courses? We believe that important benefits can be derived from teaching future students not only how to conduct themselves in an ever-growing distributed global environment but also how to think creatively and come up with innovative ideas.

References

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

[2] H. Jaakkola, J. Henno, and I. J. Rudas. “IT Curriculum as a Complex Emerging Process”. In: *2006 IEEE International Conference on Computational Cybernetics*. Aug. 2006, pp. 1–5.

[3] M. Bass. “Software Engineering Education in the New World: What Needs to Change?” In: *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*. Apr. 2016, pp. 213–221.

[4] J. Bolinger et al. “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*. Apr. 2010, pp. 1–21.

[5] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. 999133. New York, NY, USA: ACM, 2013.

[6] J. Impagliazzo, A. Clear, and H. Alrumaih. “Developing an Overview of Computing/Engineering Curricula via the CC2020 Project”. In: *2018 IEEE World Engineering Education Conference (EDUNINE)*. Mar. 2018, pp. 1–4.

[7] *SIGCSE ’11: Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*. 457110. Dallas, TX, USA: ACM, 2011.

[8] K. Beckman et al. “Collaborations: closing the industry-academia gap”. In: *IEEE Software* 14.6 (Nov. 1997), pp. 49–57.

[9] Rudolf Andersen et al. “Project courses at the NTH: 20 years of experience”. In: *Conference on Software Engineering Education*. Springer. 1994, pp. 177–188.

[10] Bernd Bruegge, Stephan Krusche, and Lukas Alperowitz. “Software Engineering Project Courses with Industrial Clients”. In: *Trans. Comput. Educ.* 15.4 (Dec. 2015), 17:1–17:31.

[11] Smrithi Rekha V and V. Adinarayanan. “An Open Source approach to enhance industry preparedness of students”. In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. Sept. 2014, pp. 194–200.

[12] Kai Mindermann, Jan-Peter Ostberg, and Stefan Wagner. “Assessing iterative practical software engineering courses with play money”. In: *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM. 2016, pp. 754–755.

[13] I. O. Pappas et al. “Empowering social innovators through collaborative and experiential learning”. In: *2018 IEEE Global Engineering Education Conference (EDUCON)*. Apr. 2018, pp. 1080–1088.

[14] Sara Isabella Freitas, John Morgan, and David Gibson. “Will MOOCs transform learning and teaching in higher education? Engagement and course retention in online learning provision”. In: *British Journal of Educational Technology* 46.3 (), pp. 455–471.

[15] Eray Tuzun, Hakan Erdogmus, and Izzet Gokhan Ozbilgin. “Are Computer Science and Engineering Graduates Ready for the Software Industry?: Experiences from an Industrial Student Training Program”. In: *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. ICSE-SEET ’18. Gothenburg, Sweden: ACM, 2018, pp. 68–77.

[16] Tony Clear et al. “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*. ITICSE-WGR ’15. Vilnius, Lithuania: ACM, 2015, pp. 1–39.

[17] M. Paasivaara et al. “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. May 2015, pp. 285–294.

[18] Fred Martin. “Toy Projects Considered Harmful”. In: *Commun. ACM* 49.7 (July 2006), pp. 113–116. ISSN: 0001-0782.

[19] *Pitchbook-NVCA, venture monitor 3Q,* 2017.

[20] Patrick Spieth, Hans Lundberg, and Kurt Matzler. “Business Model Innovation from an Entrepreneurial Perspective”. In: 18 (June 2014), pp. 261–265.

[21] R. Chatley and T. Field. “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)*. May 2017, pp. 117–126.

[22] Kevin Buffardi. “Tech Startup Learning Activities: A Formative Evaluation”. In: (2018).

[23] Kevin Buffardi, Colleen Robb, and David Rahn. “Tech Startups: Realistic Software Engineering Projects with Interdisciplinary Collaboration”. In: *J. Comput. Sci. Coll.* 32.4 (Apr. 2017), pp. 93–98. ISSN: 1937-4771.

[24] Bill Dafnis. “The Innovation Diffusion Paradox in Undergraduate Information Technology Student Outcomes”. In: *Proceedings of the 16th Annual Conference on Information Technology Education*. SIGITE ’15. Chicago, Illinois, USA: ACM, 2015, pp. 15–20.

[25] Kent Beck et al. “Manifesto for agile software development”. In: (2001).

[26] Rafael Chanin et al. “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*. ITiCSE 2018. Larnaca, Cyprus: ACM, 2018, pp. 266–271.

[27] Luciana Maria Azevedo Nascimento and Guilherme Horta Travassos. “Software Knowledge Registration Practices at Software Innovation Startups: Results of an Exploratory Study”. In: *Proceedings of the 31st Brazilian Symposium on Software Engineering*. SBES’17. Fortaleza, CE, Brazil: ACM, 2017, pp. 234–243.

[28] R. Chatley and T. Field. “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)*. May 2017, pp. 117–126.

[29] IEEE Computer Society, Pierre Bourque, and Richard E. Fairley. *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*. 3rd. Los Alamitos, CA, USA: IEEE Computer Society Press, 2014.

[30] N Robehmed. *What is a startup? Forbes*. 2013.

[31] AL Penenberg. “Eric Lies is a lean startup machine”. In: *Fast Company. September* 8 (2011).

[32] Kai Petersen. “Measuring and Predicting Software Productivity”. In: *Inf. Softw. Technol.* 53.4 (Apr. 2011), pp. 317–343.

[33] *Reference Full List:* https://tinyurl.com/yaqhn6z7.

[34] Kai Petersen et al. “Systematic Mapping Studies in Software Engineering”. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. EASE’08. Italy: BCS Learning & Development Ltd., 2008, pp. 68–77.

[35] Reid Holmes, Meghan Allen, and Michelle Craig. “Dimensions of Experientialism for Software Engineering Education”. In: *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. ICSE-SEET ’18. Gothenburg, Sweden: ACM, 2018, pp. 31–39.

[36] K. Buffardi. “Localized open source software projects: Exploring realism and motivation”. In: *2016 11th International Conference on Computer Science Education (ICCSE)*. Aug. 2016, pp. 382–387.

[37] Andres Neyem, Jose I. Benedetto, and Andres F. Chacon. “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*. SIGCSE ’14. Atlanta, Georgia, USA: ACM, 2014, pp. 391–396.

[38] Maria Paasivaara et al. “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*. ICSE-SEET ’18. Gothenburg, Sweden: ACM, 2018, pp. 49–57.

[39] Fabrício de Sousa Pinto and Paulo Caetano Silva. “Gamification Applied for Software Engineering Teaching-learning Process”. In: *Proceedings of the 31st Brazilian Symposium on Software Engineering*. SBES’17. Fortaleza, CE, Brazil: ACM, 2017, pp. 299–307.

[40] Shannon Duvall, Dugald Ralph Hutchings, and Robert C. Duvall. “Scrumage: A Method for Incorporating Multiple, Simultaneous Pedagogical Styles in the Classroom”. In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. SIGCSE ’18. Baltimore, Maryland, USA: ACM, 2018, pp. 928–933.

[41] D. Delgado et al. “Evolving a Project-Based Software Engineering Course: A Case Study”. In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. Nov. 2017, pp. 77–86.

[42] D. Delgado et al. “Evolving a Project-Based Software Engineering Course: A Case Study”. In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. Nov. 2017, pp. 77–86.

[43] Sarah Beecham, Tony Clear, and John Noll. “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*. ICGSE ’17. Buenos Aires, Argentina: IEEE Press, 2017, pp. 11–20.

[44] S. Beecham et al. “Preparing Tomorrow’s Software Engineers for Work in a Global Environment”. In: *IEEE Software* 34.1 (Jan. 2017), pp. 9–12.

[45] Ivana Bosnić et al. “Introducing SCRUM into a Distributed Software Development Course”. In: *Proceedings of the 2015 European Conference on Software Architecture Workshops*. ECSAW ’15. Dubrovnik, Cavtat, Croatia: ACM, 2015, 34:1–34:8.

[46] S. Heggen and M. Cody. “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)*. June 2018, pp. 32–39.

[47] N. M. Devadiga. “Software Engineering Education: Converging with the Startup Industry”. In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. Nov. 2017, pp. 192–196.

[48] Outi Sievi-Korte, Kari Systä, and Rune Hjelsvold. “Global vs. localffffdffffdffffdExperiences from a distributed software project course using agile methodologies”. In: *Frontiers in Education Conference (FIE), 2015 IEEE*. IEEE. 2015, pp. 1–8.

[49] N. M. Devadiga. “Software Engineering Education: Converging with the Startup Industry”. In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. Nov. 2017, pp. 192–196.

[50] M. C. Bastarrica, D. Perovich, and M. M. Samary. “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)*. May 2017, pp. 137–145.

[51] N. M. Devadiga. “Software Engineering Education: Converging with the Startup Industry”. In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. Nov. 2017, pp. 192–196.

[52] Ben Stephenson et al. “An Industrial Partnership Game Development Capstone Course”. In: *Proceedings of the 17th Annual Conference on Information Technology Education*. SIGITE ’16. Boston, Massachusetts, USA: ACM, 2016, pp. 136–141.

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

[54] Jan-Philipp Steghöfer et al. “Involving External Stakeholders in Project Courses”. In: *CoRR* abs/1805.01151 (2018).

[55] Sigrid Schefer-Wenzl and Igor Miladinovic. “Game Changing Mobile Learning Based Method Mix for Teaching Software Development”. In: *Proceedings of the 16th World Conference on Mobile and Contextual Learning*. mLearn 2017. Larnaca, Cyprus: ACM, 2017, 19:1–19:7.

[56] M. Kuhrmann and J. Mffffdffffdnch. “Enhancing Software Engineering Education Through Experimentation: An Experience Report”. In: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. June 2018, pp. 1–9.

[57] G. C. Diniz et al. “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)*. May 2017, pp. 36–42.

[58] Andreas Bollin et al. “Applying a maturity model during a software engineering course–How planning and task-solving processes influence the course performance”. In: *Journal of Systems and Software* 144 (2018), pp. 397–408.

[59] Alejandro Calderón, Mercedes Ruiz, and Elena Orta. “Integrating Serious Games As Learning Resources in a Software Project Management Course: The Case

of ProDec”. In: *Proceedings of the 1st International Workshop on Software Engineering Curricula for Millennials*. SECM ’17. Buenos Aires, Argentina: IEEE Press, 2017, pp. 21–27.

[60] Xiaoying Bai et al. “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*. ICSE-SEET ’18. Gothenburg, Sweden: ACM, 2018, pp. 58–67.

[61] S. Vathsavayi, O. S. Korte, and K. Systfffdfffd. “Tool Support for Planning Global Software Development Projects”. In: *2014 IEEE International Conference on Computer and Information Technology*. Sept. 2014, pp. 458–465.

[62] M. Nordio et al. “An experiment on teaching coordination in a globally distributed software engineering class”. In: *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE T)*. Apr. 2014, pp. 109–118.

[63] Dean Knudson et al. “Global Software Engineering Experience Through International Capstone Project Exchanges”. In: *Proceedings of the 13th Conference on Global Software Engineering*. ICGSE ’18. Gothenburg, Sweden: ACM, 2018, pp. 54–58.

[64] E. Venson et al. “Academy-industry collaboration and the effects of the involvement of undergraduate students in real world activities”. In: *2016 IEEE Frontiers in Education Conference (FIE)*. Oct. 2016, pp. 1–8.

[65] Craig Anslow and Frank Maurer. “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*. SIGCSE ’15. Kansas City, Missouri, USA: ACM, 2015, pp. 500–505.

[66] https : / / jcr . incites . thomsonreuters . com / JCRJournalHomeAction.action?.

[67] https://dbh.nsd.uib.no/publiseringskanaler/Forside.

[68] https://www.scimagojr.com/.

[69] Vebjørn Berg et al. “Software Startup Engineering: A Systematic Mapping Study”. In: *Journal of Systems and Software* (2018).

[70] Frank Elberzhager, Jürgen Münch, and Vi Tran Ngoc Nha. “A systematic mapping study on the combination of static and dynamic quality assurance techniques”. In: *Information and Software Technology* 54.1 (2012), pp. 1–15.