Towards Agile Integration within Higher Education: A Systematic Assessment

Aurelia Ciupe
Multimedia Systems and Applications Laboratory
Technical University of Cluj-Napoca
Str. Memorandumului 28, Cluj-Napoca 400114, Romania
Phone: 0264 401 200
aurelia.Ciupe@com.utcluj.ro

Razvan Ionescu
Multimedia Systems and Applications Laboratory
Technical University of Cluj-Napoca
Str. Memorandumului 28, Cluj-Napoca 400114, Romania
Phone: 0264 401 200

Serban Meza
Multimedia Systems and Applications Laboratory
Technical University of Cluj-Napoca
Str. Memorandumului 28, Cluj-Napoca 400114, Romania
Phone: 0264 401 200
serban.Meza@com.utcluj.ro

Bogdan Orza
Multimedia Systems and Applications Laboratory
Technical University of Cluj-Napoca
Str. Memorandumului 28, Cluj-Napoca 400114, Romania
Phone: 0264 401 200
bogdan.Orza@com.utcluj.ro

Abstract

Adopting an industry project-oriented approach in Higher Education teaching, research and management activities, under given quality metrics, has been intensively studied within Agile software development practices. The current work investigates Agile applicability and its practical implications into Higher Education, through a Systematic Assessment. The lack of a secondary study approach results in trending gaps and misalignments between new research proposals and the current state-of the art. Directions for further research and specific current developments are addressed in a structured manner, through relevant distributions of the publication fora according to Agile facets (research areas, methodologies, type of intervention and contribution). Study selection and data extraction have been performed under the Systematic Mapping protocol (question formulation, selection and filtering, mapping) offering a conclusive review on means of Agile integration over almost a decade.

Keywords: Agile Software Development, higher education, industry-oriented practices, systematic assessment

1. Context and Motivation

Agile software development practices have been intensively studied in the context of process improvement for complex projects. Defining the concept of Value within industry case studies stands as a critical success factor for Agile Software Intensive Development (products or services (Alahyari, Berntsson Svensson, & Gorschek, 2017). Quality (Perceived Quality, Actual Quality) and Cost have been identified as averagely Value Aspects of priority within relevant fields (Telecom, Automotive, Consultancy, Defense). Adopting and integrating industry-based practices into educational frameworks is nonetheless a future perspective of creating learning ecosystems. Higher Education Institutions (HEI), or Universities, have become an attractive environment for process and R&D operation optimization. Two directions support Agile integration into educational scenarios:

Defining Agility from the Problem-based Learning Perspective. Educational landscapes are transitioning towards community-oriented "ecological spaces", where formal, informal and life-long learning converge (Kek & Huijser, 2017). 3 fundamental cognitive skills, likewise promoted by the Partnership for 21St Century Skills (Partnership For 21St Century Skills, 2009), can be depicted as being specific to the 21st century: a) structuring large amounts of information and converting to meaningful knowledge b) critically selecting and manipulating information c) associating information to contexts, or creating new context for information usage or repositioning (Kek & Huijser, 2017). Such a framework defines the reasoning for introducing a problem-based perspective into both teaching and learning (PBL) approaches, under 9 levels (Savin-baden, 2014): PBL for knowledge management, PBL through activity, project-led PBL, PBL for practical capabilities, PBL for design-based learning, PBL for critical understanding, PBL for multimodal reasoning, collaborative distributed PBL, PBL for transformation and social reform. Following Bronfenebrenner's human ecology model (Bronfenbrenner, 2001) applied to educational contexts, the 4 corresponding concentric systems defined from a levelized organizational architecture, facilitate the learning process when transitioning between levels. Such a "fluidity" is described as a fresh perspective of adopting the model in the PBL context, under the agility umbrella (Kek & Huijser, 2017). Although the transition is mostly questionable, the ability to adapt in a flexible manner, to changes, contexts and working environments under uncertainty, is a required asset nowadays.

Integrating a project-oriented approach in both management and research activities, where quality metrics are defined and require to be assured. There is a visible project-oriented approach in both management and research activities, where quality metrics are defined and required to be assured. Mostly, injecting project management methodologies into HEI mechanisms are regarded as improvement techniques (Philbin, 2017b) at both levels: a) supporting quality assurance for educational programs b) delivering efficiency and effectiveness of operations at management level. Due to the risk imposed by the rapid growth and changes in technology, technical-profiled HEIs become even more appealing landscapes for integration. 3 core capabilities are depicted in (Philbin, 2017b), as areas of intervention: (education, knowledge exchange, research) and 4 directions of intervention: efficiency, effectiveness, economy and ethical considerations; but such a typology requires to be further extended.

The current work proposes an investigation on Agile implications into Higher Engineering Education. As Agile maps an extensive taxonomy, a structured approach needs to be considered for proposing a comparative assessment and gap identification for future research. In the context of secondary studies, Systematic Mapping has been a dedicated framework for structuring evidence in the field of Software Engineering and Agile Software Development. Section 2 presents several case studies of adopting Agile in education and the current state-of-the art of Agile Systematic Studies in Software Engineering. Section 3 describes the adopted research protocol and the selection process. Results have been discussed in Section 4, based on relevant distributions. A further perspective on the study is offered in Section 5.

2. Background and Related Studies

Several methods can be performed to achieve a secondary study based on primary studies, through structured processes, according to the main body of the research framework proposed by (Kitchenham & Charters, 2007a). Applied methods are defined in the review protocol (Kitchenham & Charters, 2007a): 1) Selection of primary studies is conducted through exclusion and inclusion criteria applied to search strategies, delivering a form of validation and relevance to the assessment (Petersen, Feldt, Mujtaba, & Mattsson, 2008), (Kitchenham & Charters, 2007a). 2) Quality assessment of primary studies is accomplished through quality instruments, assessment checklists and procedures 3) Data extraction for obtaining relevant information from primary studies based on extraction strategies (comparison, cross-checking, checking, classification) is controlled thorugh a validation process, that sometimes involves manipulation, assumptions and inferences 4) Synthesis of the extracted data is defined by a synthesis strategy (collating, summarizing, classifying, descriptive synthesis with a quantitative summary, quantitative synthesis with meta-analysis), specifying if a meta-analysis will be carried on and under which circumstances. Results of the data synthetization is represented as graphical diagrams or plots (Kitchenham & Charters, 2007a).

Systematic Literature Reviews (SLR) and Mapping Studies (MS) are 2 well defined frameworks for performing structured reviews on primary studies. While SLRs are applied to identify, evaluate, and compare all available research to answer a specific research question, a MS intends to 'map out' the research undertaken, aiming to gather research related to a specific topic (da Silva et al., 2011), rather than to answer detailed research questions. To enhance the validity of the analysis, in the case of poor quantitative experiments, a meta-analysis (MA) is performed from SLR evidence (da Silva et al., 2011). Although primarly related to biological or social sciences, systematic classification methods have been afterwards successfully applied to building classification schemes in computing and information sciences (Dwivedi, Mustafee, Williams, & Lal, 2009) (Barki, 1988). One particularly highly referenced contribution has been Barki's (Barki, 1988), who has constructed a classification scheme of Information Systems keywords, later applied in Information and Computer Science (Dwivedi, Mustafee, Williams, & Lal, 2009), and software engineering (Na, Xiaotong, Simpson, & Kim, 2004).

In software engineering, evidence exists on using SLRs, MSs and cognitive mapping to structure, organize and classify primary studies. The applicability of SLRs to the Software Engineering field has been pioneered in 2004 (Felizardo, Macdonell, Mendes, & Maldonado, 2012), as a means of systematic reviewing process. SLRs has become a research practice during the past few years, with an increase interest in developing tertiary studies. Kitchenham's (Kitchenham et al., 2009) main contribution has been to organize and analyze the existing systematic literature reviews and meta-analyses in the software engineering field (SE), given a form of tertiary study. SLR and MS studies that specifically refer to literature guidelines have reported quality enhancements (da Silva et al., 2011), although there is still a vast amount of studies that lack a quality assessment of the primary studies included in the investigation.

Although large-scale reviews are conducted under the SLR umbrella, most of the large-scale systematic assessments in Software Engineering and IT are still considered to be Mapping Studies (MS) (Turner, Kitchenham, Budgen, & Brereton, 2008). Derived from Kitchenham's guidelines (Kitchenham & Charters, 2007b), Petersen (Petersen, Feldt, Mujtaba, & Mattsson, 2008) has proposed an extension of guidelines for constructing Systematic Maps in SE, while analyzing MSs derived from existing SLRs. The latest available update on performing Systematic Mapping Studies (Petersen, Vakkalanka, & Kuzniarz, 2015) investigates the form of classification dissemination, where 6 types on visualization have been identified: heatmaps, Venn diagrams, bubble plots, bar plots, pie diagrams, line diagrams. Bubble plots (24 out of 57 selected studies) and Bar graphs (22 out of 57 selected studies) have been identified as the most used approaches. Depending on the research purpose, combined visualizations are also included to offer improved overviews on the given topics.

2.1. Systematic Assessment of Agile Practices in SE

During the latest 7 years (2010 - 2017), MSs and SLRs has been focused on technology rather than research trends. An increased complexity of topics was envisaged (49 compared to 5: 2008-2009), evidencing the development of the SE field. Highly focused topics are related to: software requirements, tests strategies, software architecture, product lines, software modeling, quality assessment and Agile practices in software development. A conclusive tertiary review (da Silva et al., 2011) questions the major SE topics investigated through SLRs, as well as the evolution of quality implications. On a basis of 67 selected studies, frequent topics have been depicted: Requirements Engineering, Distributed Software Development, Software Product Line, Software Testing, Empirical Research Methods, Software Maintenance and Evaluation, Agile Software Development.

Systematic assessments on Agile SE have been carried out on different conceptual levels. Evidence of applying the SLR framework on Agile Software Development dates since 2008, when a pioneer study has been conducted by Dyba et. al. (Dybå & Dingsøyr, 2008) to characterize several dimensions of Agile implications in SE, focusing on: agile process adoption, human and social factors, methods and their perception and related aspects (project management, SE metrics: quality, productivity satisfaction). A total number of 33 studies have been analyzed. Further on, the specificity of systematic assessments has become more granular during the upcoming 7 years, with studies addressing the entire Agile typology proposed by Dyba et al. (Dybå & Dingsøyr, 2008). User-centered Design in Agile processes in SE has been studied within a SLR carried out between 2000 – 2012 by (Salah, Paige, & Cairns, 2014). The results consisted of presenting a distribution of the 71 selected papers according to publication types (conference, journal, theses) and years. The final discussion identified 3 main categories of challenge: infrastructure, people and processes, analyzed in the context of activity parameters, work dynamics and usability testing. A more detailed implication on Agile in user-oriented software processes is analyzed through a SLR in (Brhel, Meth, Maedche, & Werder, 2015) where 83 studies are classified on 4 dimensions: process, practices, people and technology, still in accordance to the typology proposed by Dyba et al. (2008). 2 relevant studies addressing agile software requirements are both undertaken within the SLR framework (Inayat, Salim, Marczak, Daneva, & Shamshirband, 2015; Schön, Thomaschewski, & Escalona, 2017), where 20 and 30 studies have been selected after filtering as being relevant for further discussion. Combined facets on an Agile typology have been reviewed through SMs. Maps of distributions per year and domain, on quality metrics and success factors for Agile processes have been disseminated in (Rodríguez et al., 2017), where the search strategy has been constructed following a PICO strategy (Population – Intervention – Comparison – Outcomes) (Kitchenham & Charters, 2007a). A software architecture – oriented approach to Agile has been systematically described through a SM, where 54 selected studies have been visually mapped on a compound bubble chart with 3 dimensions: Agile method, year and architectural – related processes. Due to the existing matured evidence of systematic assessment in Agile technologies, associated research has recently evolved in an Agile tertiary study (Hoda, Salleh, Grundy, & Tee, 2017), where both SMs and SLRs have been collected and classified.

2.2. Applicability of Agile Practices in Education

2 specific directions of systematic studies addressing educational issues are to be depicted: tool-oriented, process-oriented. Mostly, process-oriented perspective addresses the integration of Business Process Management (BPM) into educational practices, where the tool-oriented perspective focuses on the project dimension, given a landscape that maps 3 core capabilities: Education, Research, Knowledge Exchange (Da Silva et al., 2011). The main scope of integrating Agile into HEIs, as referenced by (Philbin, 2017a), affects 4 performance factors over the core capabilities: Efficiency, Effectiveness, Economy and Ethics.

A proposal for an Agile Learning Framework is presented in (Benton & Radziwill, 2011), following 3 principles: 1) coevolution of the students and teachers in a maintained working rhythm 2) self-management and team discipline with support through communication channels 3) continuous improvement processes and individual innovation. (Angelov & de Beer, 2017) applies the characteristics of agile projects, within the context of studying how software architecture approaches might be applied to such projects in education. Focus relies on: considering distributed teams, identification of non-functional requirements, documentation and continuous development. A more-field oriented perspective is presented in (D'Souza & Rodrigues, 2015), according to which, a conceptual Agile framework (Extreme pedagogy) is provided for student-centered teaching in engineering education. Goal-oriented teaching, continuous assessment, pair learning are the 3 integrated processes derived from Agile Extreme Programming methodology, while best practices propose a student-centered approached, with specific characteristics depicted for formal meetings (that address course design and evaluation) and for informal meetings (that address the cooperation between the faculty and students for course evaluation). Learning by doing, continuous collaboration and continuous testing are the definitory characteristics in the case of a proposed Extreme Pedagogy, derived from the Xtreme Programming Three main discipline domains have been identified with relevant case studies in applying Agile methods into education: computer science (Woodward, Montgomery, Vasa, & Cain, 2013) (Tsai, Chen, & Chen, 2015), mechatronics (Edin Grimheden, 2013), business (including business informatics) (Cubric, 2013) (Zhan, Sun, & Xu, 2015). When using blended learning methods of interaction (wikis, communication channels), a significantly increased student satisfaction has been reported for knowledge improvement, teamwork skills and time-management skills. With respect to knowledge appraisal, an industry-oriented perspective derived from education, transitions the core of an Agile enterprise to a Knowledge-based Environment, where secondary and tertiary educational enrollment map the "Intelligence" feature of the Agility for the "Education" area of the Knowledge Assessment Methodology (KAM) (Trzcielinski, 2015).

2.3. Applicability of Agile Practices in Education

Most systematic assessments carried on into education, are specific to the game-based learning approaches and medical environments or practices, with a predominance of SLRs rather than MSs. Current axes of priority for systematic assessments investigate methodologies to insure educational sustainability (engineering education, management education, medical education) as well as the inclusion of new models when proposing and designing new curriculums (entrepreneurial thinking, creative thinking, reflective learning). One preliminary work analyzing educational thinking modeled in the context of transdisciplinarity, under the SLR protocol, has been presented in (Spelt, Biemans, Tobi, Luning, & Mulder, 2009). 13 studies have resulted from a set of 309 primary selected studies and classified according to 3 dimensions: potential frameworks (7 out of 13), best practices (4 out of 13), and essential conditions (2 out of 13). Although lacking the adoption of the specific SLR guidelines in terms of protocol description (database description, search strategy, visualization and mapping), the study becomes of reference in building a structured overview of a generalized topic, by referring to a specific context. A 2nd integrative literature review for higher education discusses the process of quality assessment (Gerritsen-van Leeuwenkamp, Joosten-ten Brinke, & Kester, 2017). Based on a selection of 78 journal articles, 3 themes for quality assessment have been identified (reliability, transparency, validity) and decomposed into individual subthemes, addressed under several perspectives (experts, government, staff, students). Quality assessment issues have been discussed in the process of validation and statistical data analyses, while further approaches shall be referred to each type of stakeholder.

A project-based approach within Higher Education resides on the integration of several methodologies in the form of instruments. 2 approaches have been characterized through SLRs: game-based instrumentation and simulation tools. A total number of 411 full text articles have been selected as relevant to discuss the effects of (game-based) simulations in Higher Education, under the SLR guidelines (Vlachopoulos & Makri, 2017). Most experimental designs referred to Virtual/Online games/ Simulations (88%) and general simulation games (40%), followed by computer-based learning simulations (26%), role-play games/simulations (22%), serious games (20%), while the discipline chart positions Business Marketing/Management, Computer Sciences and Science/Engineering, as the main 3 areas of applicability. The central discussion has been oriented on the improvement of the learning outcomes (affective outcomes, behavioral outcomes, cognitive outcomes), when adopting a project-oriented mindset through game/simulation-based elements. Specific to the Engineering and Computing community, the applicability of the Jigsaw technique to graduate and undergraduate studies, has been investigated under a SLR framework (Pow-sang & Escobar-c, 2017). Search strings have been constructed using a PICO strategy and a Boolean logic and have been applied to the IEEE, ACM, Scopus, Web of Science academic databases. 88 articles resulted from the search process, out of which 19 publications have been considered relevant. Computing Education and Electrical, Electronic or Telecommunications Engineering resulted to be the primary fields, with the most number of studies, followed by Chemical, Industrial, Manufacturing and General Engineering. No quality assessment has been applied.

3. Research Protocol Description

Following the Systematic Mapping guidelines provided by (Petersen, Feldt, Mujtaba, & Mattsson, 2008) and updated by (Petersen, Vakkalanka, & Kuzniarz, 2015), based on the core references documented in (Kitchenham & Charters, 2007a), the research methodology has been decomposed in 3 main phases: 1) identification of primary studies 2) selection of candidate primary studies 3) visual mapping and discussion. The goal of the current work is to 'map out' the evidences of Agile applicability in education, more precisely in the Tertiary Education. The MS protocol phases have been individually described and applied according to the MS guidelines. Figure 1 shows the main phases performed. As prerequisites, an overview of the directions that support the integration of Agile in education has been proposed for a precise understanding of the terminology, concepts, issues or challenges (namely, the Agile taxonomy).

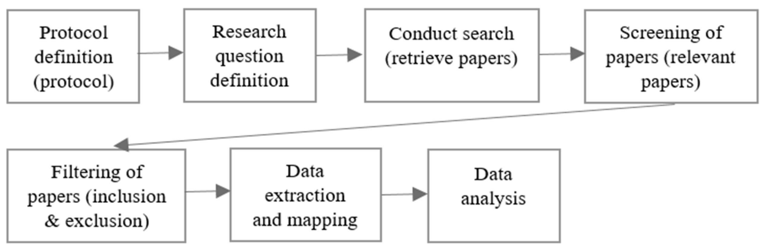


Figure 1. Depicted MS phases

3.1. Identification of primary studies

The main question that guides this mapping study and reflects its aim could be summarized as: *What are the main evidences that practical Agile applicability into higher education has become a research trend during the past 5 years*? The answer is to be approached based on 2 research question defined as research goals. A search process has been conducted to depict the set of primary studies used as input for the Systematic Mapping.

- RQ1: What is the publication fora related to practical Agile implications in education?
- RQ2: How are the Agile research areas addressed in educational fields?

Search terms have been derived from the proposed Research Questions and have been applied to the academic digital libraries to identify a bulk set of primary studies. 3 academic databases have been selected as relevant information sources : IEEE Xplore (http://ieeexplore.ieee.org/Xplore/), Science Direct (http://www.sciencedirect.com/) and Springer Link (https://link.springer.com/), based on the following premises: a) "agile implications" is a topic mainly addressed in the Software Engineering domain, where representative manuscripts have been specifically indexed by IEEE Xplore; b) studying Agile applicability in Education becomes a multi-disciplinary research direction therefore for a relevant mapping, multi-disciplinary databases need to be included c) a rigorous selection of the candidate studies requires full access to the manuscripts. All the 3 given databases allow free institutional access to the indexed manuscripts.

Search strings have been constructed on a PICO (Population – Intervention – Comparison - Outcomes) strategy, including term derivation and synonymy. Although specific to Social Sciences, updated guidelines on performing SMs in SE (Petersen, Vakkalanka, & Kuzniarz, 2015) depict specific SE terms for each category.

1. According to the guidelines, Population, refers to a role or applicability field. In the current case, higher education is targeted, therefore several levels were considered in the term taxonomy: environment, users, process

2. According to the guidelines, Intervention, refers to a methodology/ tool/ technology/ procedure. Proposed interventions have been defined according to the Agile research areas decomposed in the tertiary study provided by (Hoda, Salleh, Grundy, & Tee, 2017). Aspects related to Agile methods are identified as: method reconciliation, method analysis and method tailoring. Method tailoring applied to Agile has been studied in (Campanelli & Parreiras, 2015), under a SLR, where tailoring requires the study of a software method in the adequate context, specific in relating to the aspects, culture, objectives, environment and reality of the organization. As the PICO strategy relates the Population to the Intervention, method tailoring has been considered the practice to define the Intervention terms on 2 levels: Agile specific methods, Agile methodologies have been extracted according to the method tailoring proposed by (Campanelli & Parreiras, 2015).
3. According to the core guidelines (Kitchenham & Charters, 2007a), Comparison, defines a comparison technology. The current study compares the interventions based on the effect on the population
4. As the current study has the purpose of systemizing evidence, Outcomes have been proposed to be analyzed as mapping results.

Resulted search terms and search strings are summarized in Table 1 and Table 2. Search strings were constructed from search terms based on Boolean Logic. A progressive construction was intended to offer an overview on how the number of resulted studies decreases with search specificity.

Table 1. Search Terms

#	PICO	Related terms
1	Population	academic, education; course, class, class, student, lesson; teaching, training, learning;
2	Intervention	Agile, ASD (Agile Software Development); Scrum, Kanban, XP (Extreme Programming), Crystal, DSDM (Dynamic Systems Development Method), FDD (Feature Driven Development), RUP (Rational Unified Process, Lean; method, practice, process, approach, framework, procedure, technology, development, design, management, project, tool
3	Comparison	applicability, performance, use, functionality

3 levels of granularity have been used for search string construction (Table 2) based on the PICO strategy so that search results could be depicted for each level of complexity (Table 3). Search scoping has been observed in the case of applying all the 3 PICO levels (PIC).

Table 2. Search Strings

#	Search Strings
1	((academ* OR educati* OR universit* OR training OR learn* OR teach* OR course OR class OR student) AND (Agile))
2	((academ* OR educati* OR universit* OR training OR learn* OR teach* OR course OR class OR student) AND (Agile OR scrum OR kanban OR "extreme programming" OR crystal or DSDM or ASD or FDD or RUP OR lean))
3	((academ* OR educati* OR course OR class OR student OR lesson OR teach* OR training OR learn*)) AND (Agile) AND (method* OR development OR management OR practice OR project OR process OR approach OR design OR framework OR tool OR technology OR procedure) AND (performance OR applicability OR use OR usage OR functionality))

The number of studies retrieved for each academic library, for each search string (#1, #2, #3) is presented in Table 3.

Table 3. Search Results

#	Database	# 1	# 2	# 3
1	IEEE Xplore	3150	31965	798
2	Science Direct	14912	1502402	5276
3	Springer Link	28673	869107	25065

Primary studies resulted when all defined criteria have been applied (Search String 3: Population AND Intervention AND Comparison) were used in the selection phase. Fig. 2 shows the search process that involved 3 major steps: (I) the search strings from Table II were calibrated and applied in each of the 3 digital databases; (II) a manual search was performed in the main Journals and Conferences proceedings in the area, covering the years 2000 and 2017; (III) the ‘snow-balling’ process, in which the references of the identified papers have been analyzed.

3.2. Selection of candidate studies

The selection process included both manual and automatic screenings and has been applied separately on each database following the steps: 1) Duplicates removal (automatic) 2) Inclusion and exclusion criteria (automatic and manual) 3) Relevance screening (manual).

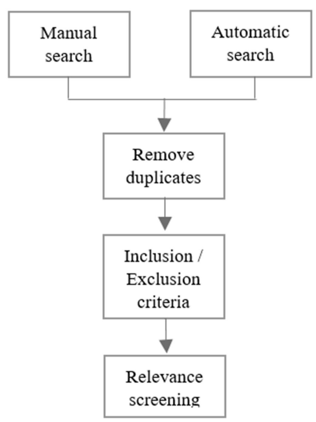


Figure 2. Depiction of the search process

1. Duplicates removal:

Duplicates have been removed on the Document Title and Author basis. As the IEEE Xplore limits the search string to 15 terms, the 3rd string has been applied in consecutive stages, on each Comparison term. Results have been aggregated and duplicates needed to be removed.

2. Inclusion and Exclusion Criteria – 1^{st} filter

The candidate studies have been selected from the initial primary studies through a metadata screening (Document Title, Publication Year, Publication Name and Abstract).

The following inclusion criteria have been applied:

- Studies published in peer-reviewed conferences, conference proceedings and journals
- Studies available in English
- Studies published after 2000, when Agile methodologies became a field of research in SE

The following exclusion criteria have been applied

- Grey literature studies, technical reports, standards, published theses, books or non-peer
- Review, opinion papers (excluding the case when student or tutor perception was discussed) and other proof-of-evidence studies (including SMs and SLRs)
- Non-full text availability

3. Relevance Screening – 2^{nd} filter

The following studies have been included:

- Studies addressing practical Agile implications in higher education (undergraduate, master, doctoral and post-graduate programs)

The following studies have been excluded:

- Studies conducted in other pedagogical environments (school, kindergarten)
- Studies conducted in organizational environments (trainings, internships, internal learning)
- Studies describing the process of teaching Agile as a discipline, with no reference to the outcomes or method tailoring and addressing agile processes in an empiric manner
- Studies modeling the Agile methodology using learning-based algorithms

A total number of 176 studies resulted from the selection process, has been mapped to address the research questions. As SMs are performed to identify literature and scientific gaps, possible implications have been discussed. While in SLRs Quality Assessment (QA) is mandatory and results mostly from subjective classifications, in the case of SMs, QA represents an optional phase, as SMs mostly focus on studies classification and mapping.

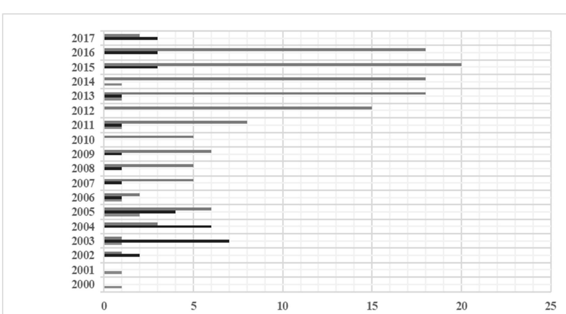
3.3. Discussion and mapping

RQ1: What is the publication fora related to practical Agile implications in education?

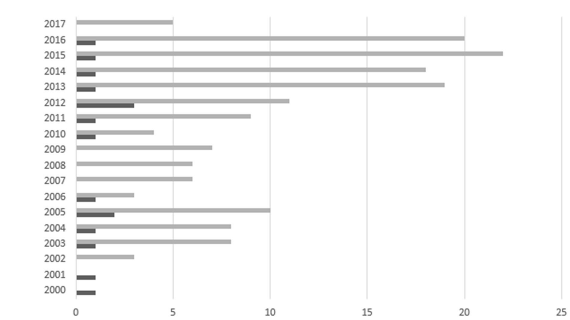
The publication continuity on the topic has been synthetized following 3 distributions: a yearly distribution of publications between 2000 - 2017 for each database (Fig.3a); a yearly distribution of publications between 2000 - 2017 on major research venues (Fig.3b) and specific publication venue identification (Figure 4).

There is an accentuated increase in the publication frequency on the topic during the past 5 years (2012 - mid 2017), with 103 studies out of a total of 176 studies (58%), distributed as: 91 studies in IEEE Xplore (88.4%), 10 studies in Springer Link (9.7%), 2 studies in Science Direct (1.9%). Even for the case of extending the entire interval to 2000 - 2017, IEEE Xplore may be referred as the main publisher with 132 studies (75.5% of the total studies), except the case of 2003, when 7 studies have been indexed by Springer Link compared to 1 study indexed by IEEE Xplore. Springer Link becomes the 2nd database for indexation, with a total number of 34 studies (19.3%), while Science Direct results to be the database with the smallest number of total publications (9 studies – 5.1%). The number of documents indexed in the years 2016 and 2015, have been maintained approximately constant in both cases: IEEE Xplore (~ 18 studies) and Springer Link (~ 18 studies).

The distribution for the past 5 years has been mapped according to the venue as: 7 journal papers (6.8% of the total number of studies 2012 – mid 2017) and conference publications 95 (92.2% of the total number of studies 2012 – mid 2017). Considering the entire interval between (2000 – 2017) conference publications are still the main source of information on the topic, with 159 studies (90%) and 17 journal papers (10%)



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
IEEE Xplore	0	0	1	1	3	6	2	5	5	6	5	8	15	18	18	20	18	2
Springer Link	0	0	2	7	6	4	1	1	1	1	0	1	0	1	0	3	3	3
Science Direct	1	1	0	1	0	2	1	0	0	0	0	1	0	1	1	0	0	0



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Conference	0	0	3	8	8	10	3	6	6	7	4	9	11	19	18	22	20	5
Journal	1	1	0	1	1	2	1	0	0	0	1	1	3	1	1	1	1	0

Figure 3. Distribution databases – years and venue type - year

A depiction of the specific publication venues is presented in Figure 4. Venues with at least 2 publications (until mid 2017) have been mapped. The most popular conference venues include: International Conference on Software Engineering (ICSE) – 15 studies (8.52% out of 176 studies), International Conference on Information Technology: New Generations (ITNG) – 11 studies (6.25 % out of 176 studies), Frontiers in Education Conference (FIE) – 10 studies (5.68% out of 176 studies).

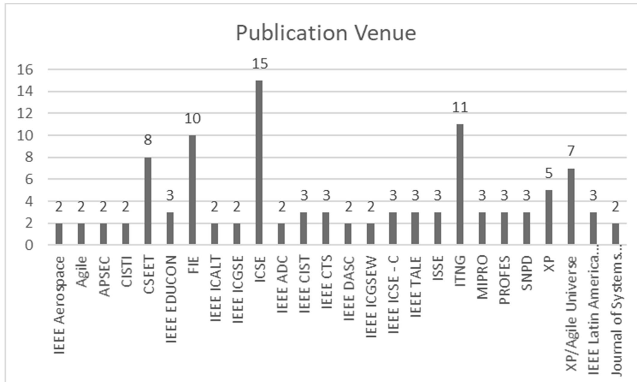


Figure 4. Distribution of publication venues

3 other software engineering – oriented venues (1 specifically education, 2 specifically Agile - oriented) become secondary representatives for the given distribution: Conference on Software Engineering Education & Training (CSEET) – 8 articles (4.54 % out of 176 studies); Extreme Programming and Agile Methods (XP/Agile Universe) – 7 articles (3.97% out of 176 studies); Agile Processes in Software Engineering and Extreme Programming (XP) – 5 articles (2.84 % out of 176 studies). 2 journal venues have been identified with more than 1 published study: IEEE Latin America Transactions (3 studies), Journal of Systems and Software (2 studies). 106 studies have been included in the distribution, while 70 remaining studies correspond to venues with less than 2 published studies.

RQ2: How are the Agile research areas addressed in educational fields?

The 2nd RQ (RQ) has been addressed through distributions following the topic multidisciplinarity, the research framework and the research areas within the Agile Methodology. The following mapping addresses the multidisciplinary facet of the research topic. 12 research fields have been identified based on the publication context (Fig. 5) : Education - E (technological aspects in Education); Agile Software Development- A (Agile Manifesto implications); Software Engineering - SE; Information Science – IS (Computers, Information Technology, Information Security); Industrial Applicability – IA (Aerospace/Mechanical/ Electrical/ Materials Engineering, Industrial Informatics, Automation); Operations – O (Control, Optimization); Communications – C; Healthcare – H (technological implications in Healthcare); Signal Processing – SP; Multimedia Technologies – MM; Human Factors – HF; Mixed – M (mixed technical topics). 90.3% of the selected studies have been published in Conference Proceedings publications (159), while 9.7% in Journals (17 studies). According to the distribution, Agile in Education has been addressed in publications on Education (45 studies – 25.5%), Agile (25 studies – 14.2%), Software Engineering (36 studies - 20.4%), Information Science (32 studies - 18.1%).

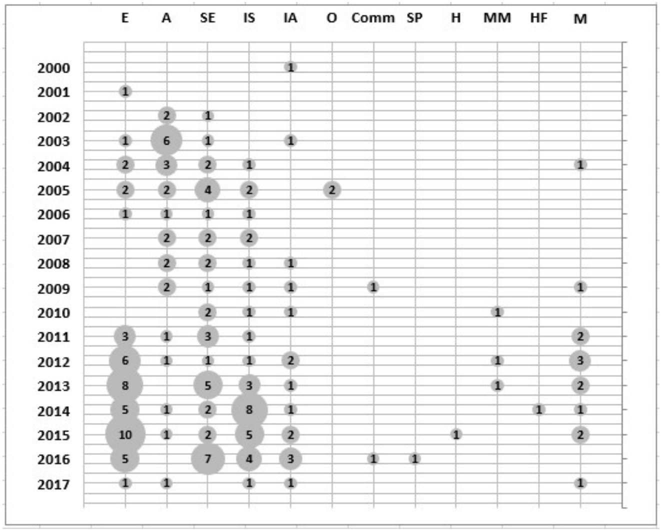


Figure 5. Distribution Distribution research field – years

The popularity of addressing the topic through the SE perspective is related to the applicability in SE case studies, where SE itself is a popular discipline in technical-oriented Higher Education. There is also evidence of case studies focused on applying Agile in Engineering Education, where from 15 associated studies (8 %), most present fields are: Electronics (4 studies) and Aerospace Engineering (4 studies) and Mechatronics (2 studies).

For an in-depth analysis, a classification scheme has been extracted from 2 recent Agile Software Development (ASD) SLRs: SLR on method tailoring (Campanelli & Parreiras, 2015) and an ASD (Agile Software Development) tertiary study (Hoda, Salleh, Grundy, & Tee, 2017). Based on the engineering cycle, Wieringa et al. introduced a classification scheme for paper evaluation in requirements engineering (Wieringa, Maiden, Mead, & Rolland, 2006): evaluation research (ER), solution proposal (SP), philosophical papers (PP) – including opinion papers, experience papers (EP), validation research (VR). ER and VR studies have been identified as the fundamental types of studies, frequently applied in software engineering (Petersen, Vakkalanka, & Kuzniarz, 2015), where instances of ER include: industrial case studies, controlled experiments with practitioners, practitioner targeted surveys, action research and ethnography studies. Simulations, laboratory experiments, prototyping, mathematical analysis and proof-of-concept, academic case studies become representative for VR.

The classification scheme has been applied to the filtered set of studies. For each study, the abstract has been considered. A depiction for each category is presented in Table 4. The N/A (Not Assigned) category refers to study that could not be mapped to any specific category.

Table 4. Research Type

#	Research Type	Studies
1	EP	S[4]; S[11]; S[12]; S[15]; S[16]; S[17]; S[18]; S[25]; S[26]; S[29]; S[30]; S[32]; S[41]; S[43]; S[54]; S[55]; S[57]; S[61]; S[66]; S[69]; S[71]; S[74]; S[79]; S[83]; S[85]; S[89]; S[90]; S[91]; S[92]; S[93]; S[94]; S[95]; S[101]; S[103]; S[109]; S[111]; S[112]; S[115]; S[116]; S[120]; S[122]; S[123]; S[125]; S[129]; S[131]; S[132]; S[133]; S[140]; S[146]; S[150]; S[152]; S[154]; S[155]; S[157]; S[164]; S[167]; S[169]; S[170]; S[172]; S[173]; S[174]; S[175]
2	ER	S[1]; S[2]; S[5]; S[7]; S[8]; S[9]; S[14]; S[33]; S[35]; S[37]; S[39]; S[40]; S[42]; S[44]; S[45]; S[46]; S[48]; S[50]; S[51]; S[52]; S[53]; S[56]; S[62]; S[65]; S[68]; S[70]; S[72]; S[73]; S[80]; S[81]; S[82]; S[96]; S[98]; S[99]; S[100]; S[102]; S[104]; S[107]; S[110]; S[114]; S[117]; S[121]; S[127]; S[130]; S[134]; S[135]; S[136]; S[137]; S[138]; S[141];

		S[142]; S[143]; S[144]; S[145]; S[147]; S[148]; S[149]; S[151]; S[153]; S[158]; S[159]; S[160]; S[162]; S[163]; S[165]; S[166]; S[170]; S[171]; S[173]; S[176];
3	PP	S[12]; S[19]; S[24]; S[27]; S[28]; S[34]; S[118]; S[139];
4	SP	S[6]; S[10]; S[13]; S[20]; S[60]; S[63]; S[75]; S[81]; S[84]; S[86]; S[87]; S[106]; S[113]; S[119]; S[124]; S[129]; S[135]; S[157];
5	VR	S[3]; S[21]; S[22]; S[23]; S[31]; S[36]; S[38]; S[47]; S[49]; S[50]; S[59]; S[64]; S[77]; S[88]; S[97]; S[105]; S[128]; S[156]; S[161]; S[168];
6	N/A	S[58]; S[67]; S[76]; S[78]; S[108]; S[126];

Given the yearly distribution for the 2000 – mid 2017 interval (Figure 6), EP and ER result to be the preferred form of publication to conduct research on Agile applicability in education, with 62 studies (33,7%) and 70 studies (38%), while less present frameworks are VR – 20 studies, SP – 7 studies, PP – 8 studies. These are still studies which cannot be identified as belonging to one specific research class (Not Assigned NA – 6 studies). There has been a growing interest in adopting the SP and VR framework during the past 6 years (2012 – 2017), with SP – 13 studies and VR – 13 studies. 8 studies have been found to adopt 2 simultaneous frameworks and have been accounted for both categories to which they belonged, increasing the number of EP (4 studies), ER (3 studies), VR (1 study).

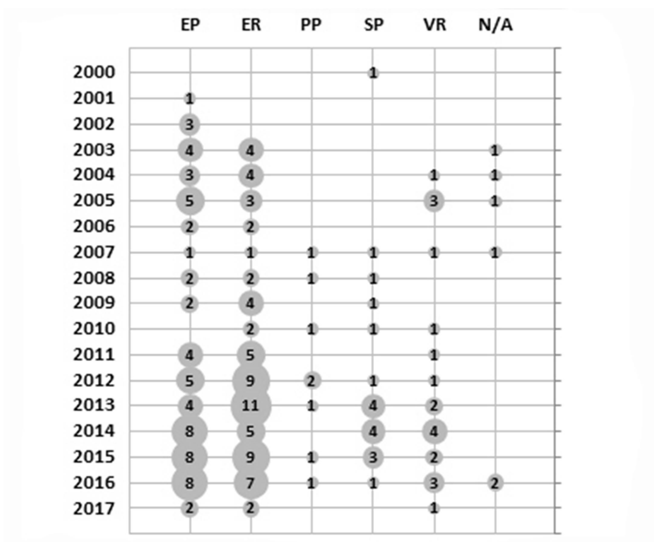


Figure 6. Distribution research classification – years.

Of interest to the current study in addressing RQ2, has been to provide evidence on the Agile research areas of priority within Higher Education and the means of evaluation. Each abstract has been scanned according to the classification criteria: Agile research area Agile methodology. In case no specific referral to the classification scheme, N/A (Not Assigned) values were associated.

Cases where studies addressed multiple classification terms for the same criterion have also been identified and associated to multiple categories. Studies addressing a general research area or including Agile methodologies as a general umbrella, have been labelled as "gen".

Due to the massive scanned content, research areas have been considered relevant and depicted from (Hoda, Salleh, Grundy, & Tee, 2017): Agile adoption (AA), Agile human and social aspects (AHS), Agile methods (AH), Agile practices (AP), Agile and Capability Maturity Model Integration (ACMMI), Agile and Usability/ User Experience (AUX), Agile and global software engineering (AGSE), Agile and the organization (AO), Agile and embedded systems (AES), Agile product line engineering (APL). The specific mapping for each Agile research area is summarized in Table 5.

Table 5. Agile Research Areas

#	Research Type	Studies
1	AA	S[12]; S[15]; S[19]; S[20]; S[21]; S[93]; S[115]; S[125]; S[73]; S[94]; S[170]; S[16];
2	ACMMI	S[34];
3	AES	S[31]; S[42]; S[44]; S[61]; S[32]; S[45];
4	AGSE	S[2]; S[3]; S[83]; S[101]; S[138]; S[152]; S[163]; S[11]; S[96];
5	AHS	S[43]; S[54]; S[62]; S[66]; S[85]; S[95]; S[99]; S[100]; S[102]; S[103]; S[159]; S[165]; S[166]; S[173]; S[175]; S[83]; S[101]; S[68]; S[89]; S[107]; S[123]; S[128]; S[53]; S[73]; S[84]; S[115]; S[129]; S[121];
6	AM	S[7]; S[14]; S[16]; S[18]; S[26]; S[37]; S[39]; S[45]; S[46]; [47]; S[48]; S[49]; S[50]; S[51]; S[55]; S[56]; S[57]; S[61]; S[64]; S[68]; S[72]; S[73]; S[75]; S[77]; S[79]; S[88]; S[106]; S[112]; S[129]; S[130]; S[134]; S[151]; S[161]; S[162]; S[171]; S[172]; S[15]; S[42]; S[163]; S[43]; S[62]; S[165]; S[25]; S[52]; S[87]; S[98]; S[104]; S[124]; S[127]; S[141]; S[144]; S[147]; S[148]; S[156]; S[120]; S[170]; S[102]; S[12]; S[85];
7	AO	S[35]; S[71]; S[121]; S[66]; S[99]; S[102]; S[51]; S[8]; S[41]; S[110]; S[84];
8	AP	S[1]; S[4]; S[5]; S[8]; S[9]; S[11]; S[17]; S[22]; S[23]; S[24]; S[25]; S[27]; S[28]; S[29]; S[30]; S[32]; S[33]; S[36]; S[38]; S[41]; S[52]; S[59]; S[69]; S[74]; S[80]; S[82]; S[87]; S[89]; S[90]; S[91]; S[92]; S[94]; S[96]; S[97]; S[98]; S[104]; S[105]; S[107]; S[109]; S[110]; S[111]; S[113]; S[114]; S[117]; S[118]; S[122]; S[123]; S[124]; S[127]; S[128]; S[131]; S[132]; S[133]; S[136]; S[137]; S[140]; S[141]; S[142]; S[143]; S[144]; S[145]; S[146]; S[147]; S[148]; S[149]; S[150]; S[154]; S[155]; S[156]; S[158]; S[160]; S[164]; S[167]; S[168]; S[170]; S[174]; S[176]; S[12]; S[19]; S[21]; S[93]; S[115]; S[125]; S[31]; S[44]; S[2]; S[3]; S[85]; S[95]; S[103]; S[159]; S[166]; S[175]; S[16]; S[18]; S[45]; S[50]; S[57]; S[64]; S[77]; S[79]; S[106]; S[112]; S[129]; S[171]; S[172]; S[35]; S[71]; S[121]; S[86]; S[116]; S[139]; S[169]; S[83]; S[163]; S[165]; S[66]; S[51];
9	APL	S[14];
10	AUX	S[53]; S[70];S[167]
11	Tools	S[6]; S[10]; S[13]; S[20]; S[40]; S[60]; S[63]; S[65]; S[81]; S[84]; S[86]; S[116]; S[119]; S[120]; S[135]; S[139]; S[157]; S[169]; S[138]; S[75]; S[151]; S[123];S[124];
12	Gen	S[153];
13	N/A	S[58]; S[67]; S[76]; S[78]; S[108]; S[126];

Agile methodologies have been extracted for the following mapping. Most of the studies adopt Agile as a general set of guidelines (93 studies). SCRUM (40 studies) and XP (38 studies) are the most present Agile methodologies. With respect to the XP methodology, there is a specific interest in including Pair Programming (PP) practices in both course and practical activities (9 studies). Other methodologies refer to: Lean (6 studies), Test-driven development TDD (6 studies), Model-driven development MDD (1 study), Rational Unified Process RUP (2 studies). Most of the studies have been carried out as ER (75 studies) and EP (67 studies). Studies mapping into each category is presented in Table 6.

Table 6. Agile Methodologies

#	Research Type	Studies
1	Agile Scrum	s[1], s[23], s[25], s[31], s[35], s[37], s[46], s[50], s[51], s[60], s[61], s[63], s[72], s[73], s[74], s[80], s[84], s[85], s[87], s[96], s[104], s[105], s[113], s[115], s[122], s[123], s[124], s[128], s[133], s[134], s[135], s[138], s[139], s[140], s[161], s[162], s[163], s[169], s[170], s[171]
2	XP	s[2], s[7], s[15], s[16], s[17], s[18], s[26], s[29], s[39], s[49], s[52], s[53], s[56], s[60], s[65], s[77], s[79], s[80], s[83], s[88], s[93], s[95], s[96], s[99], s[102], s[103], s[106], s[112], s[127], s[130], s[136], s[142], s[148], s[154], s[157], s[158], s[160]
3	Pair Programming	s[53], s[55], s[68], s[95], s[110], s[144], s[147], s[159], s[175]

4	Lean	s[12] , s[14], s[98], s[168]
5	Test-driven development	s[42], s[47], s[57], s[75], s[109], s[147]
6	Model-driven development	s[45]
7	Rational Unified Process	s[130], s[159]
8	N/A	s[58], s[67], s[78], s[108], s[126]

Two cumulative mappings are proposed to illustrate how agile methodologies address Agile research areas and under which framework (research framework), respectively the applicability to research domains (Figure 7) In case of ERs, methodologies have been proposed to be evaluated in specific educational contexts while in the case of EPs, case studies and lessons learnt have been presented. VR (21 studies) demonstrate the impact, mainly positive, of the Agile methodologies. New solutions (SP) in terms of extended Agile methodologies or practical tools have also been proposed (18 studies).

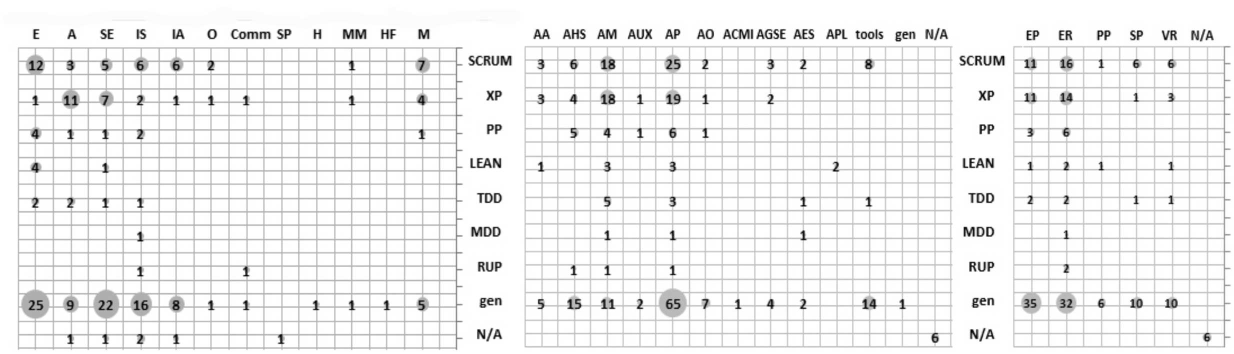


Figure 7. Distribution Agile methodology – Agile research area – research type / research domain

As seen in Figure 7, selected studies address Agile from the practice integration perspective (AP – 123 studies) and method applicability (AM – 61 studies) compared to the tertiary review where second dominant research area was Usability (AUX). SCRUM and XP practices are applied in contexts varying from: curricula design to course delivery, project requirements gathering and project evaluation and tracking. Human factors and tools result to be the following criteria of interest: 31 studies, 23 studies. As the distribution shows, there is a specific focus on observing the team dynamics in educational environments (participation, implication, impact, etc) as well as proposing supporting Agile tools (platforms, applications).

With respect to the applicability of Agile methodologies to academic domains, several aspects could be emphasized. Agile SCRUM is the main Agile methodology used with (40 studies), followed closely by XP (38 studies). Pair Programming is the 3rd methodology in the classification, accounting much less studies (9 studies). The other categories like TDD, MDD and RUP are less present, with 6, 1 and 2 studies each. However, most of the 176 studies are included in the Agile general umbrella category, "gen" (93 studies).

4. Conclusions

Given the latest trends of conducting systematic assements on Agile (Agile methods tailoring, tertiary studies etc), the current work gathers evidence on Agile implications in education through a systematic mapping study, following 2 research questions, one presenting an in-depth view on the publication fora according to the indexing database and publication venues. The yearly evolution according to educational fields is presented together with a study distribution on academic databases, to create the general context of publication plethora on the topic. Secondly, studies have been classified according to a specific scheme methodology – research type – research areas to present in a quantitative manner: through which means of research, which Agile research areas have been studied through which Agile methodologies. In terms of plotting, bubble charts and bar graphs were preferred for mapping and visualization, according to given guidelines in performing systematic mappings. Resulted mappings offer an insight on the current state-of-the art and possible areas of interventions for future studies. Apart from the given classification scheme and discussion, 2 directions arise for future research: game-based Agile practices (5 studies), Agile in a PBL context (3 studies). Future extensions propose to extend the current mapping to a SLR where quality assessment becomes a mandatory phase of the protocol.

References

Alahyari, H., Berntsson Svensson, R., & Gorschek, T. (2017). *A study of value in agile software development organizations*. Journal of Systems and Software, 125, 271–288.

Angelov, S., & de Beer, P. (2017). *Designing and applying an approach to software architecting in agile projects in education*. Journal of Systems and Software, 127, 78–90.

Barki, Henri and Rivard, Suzanne. (2008). *An Information Systems Keyword Classification Scheme*. MIS Quarterly, 12:2

Benton, M. C., & Radziwill, N. M. (2011). *A path for exploring the agile organizing framework in technology education*. Proceedings - 2011 Agile Conference, Agile 2011, 131–134.

Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). *Exploring principles of user-centered agile software development: A literature review*. Information and Software Technology, 61(0), 163–181.

Bronfenbrenner, U. (2001). *The Bioecological Theory of Human Development. Making Human Beings Human*: Bioecological Perspecrtives on Human Development.

Campanelli, A. S., & Parreiras, F. S. (2015). *Agile methods tailoring - A systematic literature review*. Journal of Systems and Software, 110, 85–100.

Cubric, M. (2013). *An agile method for teaching agile in business schools.* The International Journal of Management Education, 11(3), 119–131.

D'Souza, M. J., & Rodrigues, P. (2015). *Extreme pedagogy: An agile teaching-learning methodology for engineering education*. Indian Journal of Science and Technology, 8(9), 828–833.

Da Silva, F. Q. B., Santos, A. L. M., Soares, S., Frana, A. C. C., Monteiro, C. V. F., & MacIel, F. F. (2011). *Six years of systematic literature reviews in software engineering: An updated tertiary study*. Information and Software Technology, 53(9), 899–913.

da Silva, F. Q. B., Santos, A. L. M., Soares, S., França, A. C. C., Monteiro, C. V. F., & Maciel, F. F. (2011). *Six years of systematic literature reviews in software engineering: An updated tertiary study*. Information and Software Technology, 53(9), 899–913.

Dwivedi, Y. K., Mustafee, N., Williams, M. D., & Lal, B. (2009). *Classification of information systems research revisited: A keyword analysis approach*. Proceedings - Pacific Asia Conference on Information Systems, 105.

Dybå, T., & Dingsøyr, T. (2008). *Empirical studies of agile software development: A systematic review*. Information and Software Technology, 50, 833–859.

Edin Grimheden, M. (2013). *Can agile methods enhance mechatronics design education? Mechatronics*, 23(8), 967–973.

Felizardo, K. R., Macdonell, S. G., Mendes, E., & Maldonado, J. C. (2012). *A systematic mapping on the use of visual data mining to support the conduct of systematic literature reviews.* Journal of Software, 7(2), 450–461.

Gerritsen-van Leeuwenkamp, K. J., Joosten-ten Brinke, D., & Kester, L. (2017). *Assessment quality in tertiary education: An integrative literature review*. Studies in Educational Evaluation, 55(September), 94–116.

Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2017). *Systematic literature reviews in agile software development: A tertiary study*. Information and Software Technology, 85, 1339–1351.

Kek, M. Y. C. A., & Huijser, H. (2017). *Problem-based Learning into the Future.*

Kitchenham, B., & Charters, S. (2007a). *Guidelines for performing Systematic Literature Reviews* in Software Engineering. Engineering, 2, 1051.

Kitchenham, B., & Charters, S. (2007b). *Guidelines for performing Systematic Literature Reviews* in Software Engineering.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). *Systematic literature reviews in software engineering - A systematic literature review*. Information and Software Technology, 51(1), 7–15.

Na, K. S., Xiaotong, L., Simpson, J. T., & Kim, K. Y. (2004). *Uncertainty profile and software project performance: A cross-national comparison*. Journal of Systems and Software, 70, 155–163.

Partnership For 21St Century Skills. (2009). 21st CENTURY STUDENT OUTCOMES. Framework, 1–9.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic mapping studies in software engineering*. EASE'08 Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, 68–77.

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). *Guidelines for conducting systematic mapping studies in software engineering: An update*. Information and Software Technology, 64(0), 1–18.

Philbin, S. P. (2017a). *Investigating the Scope for Agile Project Management to Be Adopted by Higher Education Institutions*. In Z. Mahmood (Ed.), Software Project Management for Distributed Computing: Life-Cycle Methods for Developing Scalable and Reliable Tools (pp. 345–366). Cham: Springer International Publishing. Philbin, S. P. (2017b). Software Project Management for Distributed Computing, 430.

Pow-sang, J. A., & Escobar-c, P. (2017). *Interactive Collaborative Robotics*, 10459.

Rodríguez, P., Haghighatkhah, A., Lwakatare, L. E., Teppola, S., Suomalainen, T., Eskeli, J., Oivo, M. (2017). *Continuous deployment of software intensive products and services: A systematic mapping study*. Journal of Systems and Software, 123, 263–291.

Salah, D., Paige, R. F., & Cairns, P. (2014). *A Systematic Literature Review for Agile Development Processes and User Centred Design Integration*. Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14), London, United Kingdom, 13-14 May, 2014, 5:1--5:10.

Savin-baden, M. (2014). *Using Problem-Based Learning : New Constellations for the 21 st Century*. Journal on Excellence in College Teaching, 25, 197–219.

Schön, E.-M., Thomaschewski, J., & Escalona, M. J. (2017). *Agile Requirements Engineering: A systematic literature review*. Computer Standards & Interfaces, 49, 79–91.

Spelt, E. J. H., Biemans, H. J. A., Tobi, H., Luning, P. A., & Mulder, M. (2009). *Teaching and learning in interdisciplinary higher education: A systematic review*. Educational Psychology Review, 21(4), 365–378.

Trzcielinski, S. (2015). *The Influence of Knowledge Based Economy on Agility of Enterprise*. Procedia Manufacturing, 3(Ahfe), 6615–6623.

Tsai, W. L., Chen, C. Y., & Chen, C. S. (2015). *Snowman: Agile development method with institutionalized communication and documentation for capstone projects*. Asia Pacific Management Review, 23(1), 12–19.

Turner, M., Kitchenham, B., Budgen, D., & Brereton, O. P. (2008). *Lessons Learnt Undertaking a Large-Scale Systematic Literature Review*. Proceedings of EASE, 110–118.

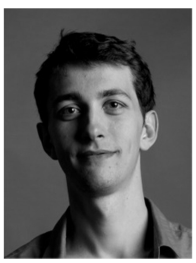
Vlachopoulos, D., & Makri, A. (2017). *The effect of games and simulations on higher education: a systematic literature review*. International Journal of Educational Technology in Higher Education (Vol. 14). International Journal of Educational Technology in Higher Education.

Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). *Requirements engineering paper classification and evaluation criteria: A proposal and a discussion*. Requirements Engineering, 11, 102–107.

Woodward, C. J., Montgomery, J., Vasa, R., & Cain, A. (2013). *Agile development spikes applied to computer science education*. Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), (August), 699–704.
Zhan, D., Sun, C., & Xu, X. (2015). Software Engineering Education (Vol. 327).

Aurelia Ciupe owns a BSc in Electronics and Telecommunications Engineering, MSc in Telecommunications Technology and Systems, of the Technical University of Cluj-Napoca. She has been involved in several POSDRU projects, with job positions varying from: class tutor and curriculum creator, to LMS software developer. She currently holds a teaching assistant position in the Multimedia Systems and Applications Laboratory, of Technical University of Cluj-Napoca, with activities in: applied informatics, multimedia technologies and content management systems.

Razvan Ionescu is currently undergoing his M.Sc studies at the Faculty of Elecronics, Telecoominications and Information Technology of TUCN, in the field of Multimedia Technolgies. He is actively involved in professional activities and practical projects within the Software Development and Quality Assurance Industry. His research interests include applied project management methodologies and process analysis in the field of Continuous Delivery.



Serban-Nicolae Meza owns a PhD in video signal processing from the Technical University of Cluj-Napoca. Currently he is with the Multimedia Technologies and Distant Education Centre from the same university, integrating his passion for video and multimedia systems into research and developments for the industry. He has been actively involved in Romanian (national scale) educational projects like www.didatec.ro, www.e-start.ro, www.e-administratie.forhe.ro and colaborated with companies like Grass Valley – Netherlands in the manufacturing of high-end professional video cameras. He is interested in socially-interactive educational platforms and 3D augmented reality video.

Bogdan Orza focuses both at education and research level on information management, video and television engineering, digital image processing, multimedia systems and applications, eLearning solutions, database design and programming. He acts as the coordinator of the MSAL group and the director of the TUCN's Department of Continuing Education and Distance Learning, being actively involved in national and regional coordination of numerous educational projects.

Reference Material

#	Document Title	#	Document Title
1	A Capstone Course on Agile Software Development Using Scrum	89	Improve the accomplishment of project in PBL learning by using an agile method
2	A Case Study in the Use of Extreme Programming in an Academic Environment	90	Improvements for agile manifesto and make agile applicable for undergraduate research projects
3	A Crash Undergraduate Course in Global Software Engineering	91	Innovation-directed experiential learning using service blueprints
4	A cross-program investigation of students' perceptions of agile methods	92	Interdisciplinarity in Computer Science: A Case Study on Graduate Courses
5	A Cyber Security Multi Agency Collaboration for Rapid Response that Uses AGILE Methods on an Education Infrastructure	93	Introducing agile into a software development Capstone project
6	A methodology for building simulation-based e-learning environments for Scrum	94	Introducing Agile Methods in Learning Environments: Lessons Learned
7	A Methodology for Computer Programming Teaching Based on Bloom's Taxonomy of Educational Objectives and Apllied Through the Pair Programming	95	Investigating Pair Programming in a Software Engineering Course in an Asian Setting
8	A Process Model for Supporting the Management of Distance Learning Courses Through an Agile Approach	96	Issues and mitigation strategies when using agile industrial software development processes in student software engineering projects
9	A proposal of postgraduate programme for software testing specialization	97	jFakih: Modelling mobile learning game
10	A proposition in design education with a potential in commercial venture in small aircraft manufacture	98	Keeping the Spin -- From Idea to Cash in 6 Weeks: Success Story of Agile/Lean Transformation
11	A self-organized team: Managing agile content development for seminars	99	Knowledge repository to improve agile development processes learning
12	A Software Engineering Perspective for Accelerating Educational Technologies	100	Learning Agile Software Development in High School: An Investigation
13	A web based tool for teaching hardware design based on the plain simple hardware description language	101	Learning Global Agile Software Engineering Using Same-Site and Cross-Site Teams
14	Accelerating educational technologies using software product lines	102	Lessons Learned from an XP Experiment with Students: Test-First Needs More Teachings
15	Adaptations for Teaching Software Development with Extreme Programming: An Experience Report	103	Managing a project course using Extreme Programming
16	Adapting extreme programming for a core software engineering course	104	Measuring Best-in-Class Software Releases
17	Adapting XP to an Academic Environment by Phasing-In Practices	105	Measuring the Impact of Agile Coaching on Studentsâ€™ Performance
18	Agile CS1 Labs: eXtreme Programming Practices in an Introductory Programming Course	106	Methodology for developing ICT based course material for children with a developmental disability based on EPISODE
19	Agile development spikes applied to computer science education	107	Methods agile and methodology A+S in the teaching of software engineering
20	Agile engineering in mechatronic education: Interdisciplinary development of a mechatronic system using Scrum	108	On the Impact of a Collaborative Pedagogy on African American Millennial Students in Software Engineering
21	Agile hardware and software system engineering for innovation	109	On the Influence of Test-Driven Development on Software Design
22	Agile Hour: Teaching XP Skills to Students and IT Professionals	110	On the Integration of Agile Practices into Teaching: An Approach to Overcoming Teaching and Learning Challenges of Programming
23	Agile management: A teaching model based on SCRUM	111	Patterns of Practice and Design: Towards an Agile Methodology for Educational Design Research
24	Agile Manifesto in Higher Education	112	Perceptions of Agile Practices: A Student Survey
25	Agile methodology for spacecraft ground software development: A cultural shift	113	PlayScrum - A Card Game to Learn the Scrum Agile Method
26	Agile Methods in Software Engineering Education	114	Power of Recognition: A Conceptual Framework for Agile Capstone Project in Academic Environment
27	Agile project management framework	115	Rapid improvement of students' soft-skills based on an agile-process approach
28	Agile project management in product design	116	Reflections on using the Internet for the evaluation of course delivery
29	Agile Software Development Learning through Open Hardware Project	117	Research and practice on the whole life cycle test process model
30	Agile Software Development Supporting Higher Education Reform	118	Research Documentation Guidelines - Capturing knowledge, improving research
31	An Academic Case Study Using Scrum	119	RFID-Enable in the Real-Time Multi-experiment Training Center
32	An Agile embedded systems capstone course	120	Scalability of assessments of wiki-based learning experiences in higher education
33	An agile learning design method for open educational resources	121	Scrum in a software engineering course: An in-depth praxis report
34	An agile strategy for implementing CMMI project management practices in software organizations	122	Scrum solo: Software process for individual development
35	An Agile Translation Process for complex innovations: An Industry/University Cooperative Research Center case study	123	ScrumLint: Identifying Violations of Agile Practices Using Development Artifacts
36	An Approach to Software Architecting in Agile Software Development Projects in Education	124	ScrumTutor: A web-based interactive tutorial for Scrum Software development
37	An interdisciplinary academic project for spatial critical embedded system agile development	125	Security in Agile Development: Pedagogic Lessons from an Undergraduate Software Engineering Case Study
38	An Iterative and Agile Process Model for Teaching Software Engineering	126	Shaping the Afghan learning settings and power to the learners through a high-tech library management system
39	An XP Experiment with Students â€“ Setup and Problems	127	Software Development and Experimentation in an Academic Environment: The Gaudi Experience
40	AOSTA: Agile Open Source Tools Academy	128	Software Development PBL Focusing on Communication Using Scrum
41	Application of agility in development of IT-Portal for the university: A case study	129	Software Kaizen: Using Agile to Form High-Perfomance Software Development Teams
42	Applying Acceptance Test Driven Development to a Problem Based Learning Academic Real-Time System	130	Software process in the classroom: A comparative study
43	Applying Agile Method on Academic Access and Fraud Control System	131	Student experiences with executable acceptance testing
44	Applying Interdisciplinarity and Agile Methods in the Development of a Smart Grids System	132	Student-faculty collaboration on design of innovative agile software engineering curriculum
45	Applying Interdisciplinarity and Agile Methods in the Development of an Embedded System	133	Students' perceptions of Scrum practices
46	Applying Scrum in an Interdisciplinary Project Using Big Data, Internet of Things, and Credit Cards	134	Supporting assessment of practices in software engineering courses
47	Applying the TBC method in introductory programming courses	135	Supporting Virtual Meetings in Distributed Scrum Teams
48	Assertion-Driven Development: Assessing the Quality of Contracts Using Meta-Mutations	136	TaskBoard - Using XP to Implement Problem-Based Learning in an Introductory Programming Course
49	Being Extreme in the classroom: Experiences teaching XP	137	Task-centric holistic agile approach on teaching cyber physical systems engineering
50	Can agile methods enhance mechatronics design education?	138	Teaching a globally distributed project course using Scrum practices
51	Case Studies of Industry-Academia Research Collaborations for Software Development with Agile	139	Teaching Agile Collaboration Skills in the Classroom
52	Cognitive aspects in a project-based course in software engineering	140	Teaching Agile Methods to Software Engineering Professionals: 10Â Years, 1000 Release Plans
53	Collaborating on mobile app design through pair programming: A practice-oriented approach overview and expert review	141	Teaching agile software development at university level: Values, management, and craftsmanship
54	Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight	142	Teaching eXtreme Programming in a Project-Based Capstone Course
55	Collaboration leads to success: A study of the effects of using pair-programming teaching technique on student performance in a Middle Eastern society	143	Teaching for understanding and its specialization to software engineering

56	Comparing Extreme Programming to Traditional Development for Student Projects: A Case Study	144	Teaching mobile application development through lectures, interactive tutorials, and Pair Programming
57	Comparing the Defect Reduction Benefits of Code Inspection and Test-Driven Development	145	Teaching software process modeling
58	Computational agile beam ladar imaging	146	Teaching students software engineering practices for micro-teams
59	Constructivist Pedagogy Meets Agile Development Â– A Case Study	147	Teaching TDD, the Coding Dojo Style
60	Developing 3D action puzzle game application "The Mechanic" using UNITY	148	Teaching Team Work: An Extreme Week for First-Year Programmers
61	Developing a CDS with Scrum in an interdisciplinary academic project	149	Teaching teamwork in engineering and computer science
62	Developing a Smart Grids System as a PBL with Agile Method on Computer Engineering Courses	150	The agile approach in an undergraduate software engineering course project
63	Developing an Educational Medical Game Using AgilePASSI Multi-agent Methodology	151	The Agile Teaching/Learning Methodology and Its e-Learning Platform
64	Developing Entrepreneurial Skills in IT Courses: The Role of Agile Software Development Practices in Producing Successful Student Initiated Products	152	The learner experience of student-led international group project work in software engineering
65	DevOpsUse for Rapid Training of Agile Practices Within Undergraduate and Startup Communities	153	The role of a project-based capstone course
66	Distributed Software Development with One Hand Tied Behind the Back: A Course Unit to Experience the Role of Communication in GSD	154	The Software Hut â€“ A Student Experience of eXtreme Programming with Real Commercial Clients
67	e-Enterprise and management course development using strategy formulation framework for vocational education	155	The virtual agile enterprise: Making the most of a software engineering course
68	Eleven Guidelines for Implementing Pair Programming in the Classroom	156	Toward a Framework for Designing Adaptive Educational Hypermedia System Based on Agile Learning Design Approach
69	Empowering Students and the Community through Agile Software Development Service-Learning	157	Towards a flexible open-source software library for multi-layered scholarly textual studies: An Arabic case study dealing with semi-automatic language processing
70	Engineering M-Learning Using Agile User-Centered Design	158	Towards an agile project management in product design
71	Enterprise architecture of Colombian Higher Education	159	Towards experiencing the pair programming as a practice of the Rational Unified Process (RUP)
72	Errors in E-learning: the key is global approach for optimal education	160	Towards model-driven game engineering for serious educational games: Tailored use cases for game requirements
73	Evaluating capstone project through flexible and collaborative use of Scrum framework	161	Tracking the student's performance in Web-based education using Scrum methodology
74	Evaluation of software engineering management best practices in the Western Cape	162	Training scrum with gamification: Lessons learned after two teaching periods
75	Experiences Using Automated 4ests and 4est Driven Development in Computer 9cience I	163	Transitioning to Distributed Development in Students' Global Software Development Projects: The Role of Agile Methodologies and End-to-End Tooling
76	Extreme Programming in Curriculum: Experiences from Academia and Industry	164	Updating CS capstone projects to incorporate new agile methodologies used in industry
77	eXtreme Programmingâ€“â€“helpful or harmful in educating undergraduates?	165	Use of Agile Methods in Software Engineering Education
78	eXtreme Programmingâ€“â€“helpful or harmful in educating undergraduates?	166	Using a role scheme to derive software project metrics
79	Extreme Teaching â€" An Agile Approach to Education	167	Using agile management to share course projects among multiple classes
80	Filleting XP for Educational Purposes	168	Using innovation as a catalyst for integrating business and engineering education
81	Framework for web application domain knowledge extraction	169	Using Scrum and Wikis to manage student major projects
82	Generic service integration in adaptive learning experiences using \IMS\ learning design	170	Using Scrum for Software Engineering Class Projects
83	Global vs. local â€" Experiences from a distributed software project course using agile methodologies	171	Using scrum to teach software engineering: A case study
84	Helping students track learning progress using burn down charts	172	Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum
85	How Surveys, Tutors and Software Help to Assess Scrum Adoption in a Classroom Software Engineering Project	173	When Teams Go Crazy: An Environment to Experience Group Dynamics in Software Project Management Courses
86	Identification of Inaccurate Effort Estimates in Agile Software Development	174	Where in the Curriculum Is the Right Place for Teaching Agile Methods?
87	Implementation of a haptic simulation environment for surgical learning systems	175	Work in progress-unexpected student outcome from collaborative agile software development practices and paired programming in a software engineering course
88	Implementing Computer-Aided Language Learning Tool Using Hybrid Agile Method: A Case Study	176	Workshop: Learning agile through active learning activities