Audience and Content Areas of Online Software Engineering Education and Training: A Systematic Review

Kevin D. Wendt
University of Minnesota
wendt@cs.umn.edu

Abstract

Online courses and instruction are a popular and rapidly expanding medium for learning. However, no holistic summary exists which classifies who or what is being taught and how online courses are being researched in the field of software engineering education and training (SEET). More important, prior research does not cover what gaps exist in the literature: which areas of SEET being ignored altogether in research. This paper reports the results of a systematic review of the existing literature for online SEET and provides analysis of the audiences and content areas being researched. Grounded in established guidelines for systematic reviews in software engineering, the studies that are identified are then coded using three standards for software engineering education content areas. As a result of the systematic review, more than 9000 search results were analyzed. Inclusion and exclusion of studies in predetermined stages resulted in more than 30 studies being coded for audience and/or software engineering content areas. The result is a comprehensive picture of the current state of research in online SEET and an identification of the gaps to be addressed.

1. Introduction

Computer science and software engineering enrollments are on the rise across the United States [1]. But, even as they do, it is expected that the growth will fall short, leaving an ever-increasing number of job openings [2]. As these high-paying jobs remain open, more and more positions will be filled by those without the necessary education and training, by those who do not have a degree in computer science. One attractive method of gaining that education and training, without the traditional campus experience (and cost), is through online courses.

Recent reports indicate that online courses will continue to see enrollment growth [3]. With that increase, the potential for educational research grows, too. The purpose of this paper is to clarify the state of research regarding online software engineering education and training (SEET). Individual research papers identify outcomes, detail processes and tools, report on the creation of individual courses and entire online degree programs, and some suggest best practices for online SEET. To date, however, there is no holistic overview of these research findings. Specifically, no coherent analysis has been performed for what is being taught and to whom. Without a comprehensive effort to identify what is being done and who the target audiences are, we risk neglecting areas of software engineering education and training that could be provided based on demographics and industry needs. This paper provides a holistic view of the published research regarding offerings of online SEET courses. This systematic approach provides researchers and practitioners with an understanding of what has been done and what remains to be offered and investigated. Seemingly high-demand areas of SEET were found to be unreported, potentially identifying important areas yet to be serviced by this growing pedagogy.

2. Background

To accurately assess the intended software engineering skills that each study may report, a systematic approach to identifying those skills is necessary. To that end, three established standards will be used to code each relevant study identified by the systematic review: the IEEE Computer Society (IEEE-CS) Software Engineering Competency Model published in 2014, the joint IEEE-CS and Association of Computing Machinery (ACM) Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering published in 2014 and the joint IEEE-CS/ACM Curriculum Guidelines for Graduate Degree Programs in Software Engineering published in 2009.

To further ensure quality, an established process of performing systematic reviews in the field will

URI: https://hdl.handle.net/10125/60198
ISBN: 978-0-9981331-2-6

be followed: Guidelines for performing Systematic Literature Reviews in Software Engineering by Kitchenham and Charters [4].

2.1. Software Engineering Competency Model

The IEEE-CS Software Engineering Competency Model (hereafter, SWECOM) describes the working "competencies" a software engineer should exhibit [5]. These competencies are broken down by the activities, skills (sets of activities), and skill sets (sets of skills) that define the working abilities necessary to be a successful software engineer. SWECOM defines 13 skill areas for a software engineer, including standard life cycle stage expertise, such as Requirements, Design, and Testing, but also cross-cutting areas, such as Safety, Quality, and Measurement. It also defines standards that a software engineers work would need to meet in order to be considered to have met one of "five levels of increasing competency." These levels include Technician, Entry Level, Practitioner, Technical Leader and Senior Software Engineer.

2.2. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering

The Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (hereafter, SE2014) was the result of the Joint ACM/IEEE-CS task force to update the prior 2004 curriculum guidelines [6]. These guidelines details 9 knowledge areas for undergraduates. These knowledge areas are closely related to the skills areas described in SWECOM, but are organized differently. In particular, the inclusion of Professional Practice is a unique element of these guidelines.

2.3. Curriculum Guidelines for Graduate Degree Programs in Software Engineering

The Graduate Software Engineering 2009: Curriculum Guidelines for Graduate Degree Programs in Software Engineering (hereafter, GSwE2009) was the result of the work of the Integrated Software & Systems Engineering Curriculum (iSSEc) Project, published by both ACM and IEEE [7]. It details 11 knowledge areas for graduate study, including an Ethics area not found in other guidelines.

2.4. Guidelines for performing Systematic Literature Reviews in Software Engineering

The Guidelines for performing Systematic Literature Reviews in Software Engineering (hereafter, SLR-SE Guidelines) was published in 2007 by lead authors Kitchenham and Charters. [4] This Evidence-Based Software Engineering (EBSE) technical report was the result of the work of the Software Engineering Group of Keele University and the Department of Computer Science of the University of Durham, both of the UK. Based on the demonstrated need for a clear model for performing systematic literature reviews (also referred to as systematic reviews), the two lead authors, four internal reviewers from the EBSE Project (EP/CS51839/X), and three external reviewers distilled the guidance of three existing sets of systematic review guidelines, their own experiences, and a handful of textbooks to create new guidelines for the performance of systematic reviews in software engineering. Drawing heavily from the medical field's systematic review process, the SLR-SE Guidelines are "aimed primarily at software engineering researchers including PhD students." It was with this in mind that the guidelines were followed when completing the work and writing this paper.

3. Method

3.1. Research Questions

The questions answered by this paper are

- R1 What content is being taught by online SEET courses?
- R2 Who is the audience of online SEET courses?
- R3 What is the trend of research of online SEET courses, measured by studies published year over year?

3.2. Research Methodology

In order to ensure quality in the process of this systematic review, this paper follows the process set forth in the SLR-SE Guidelines[4]. The results of following the recommended procedure can be found in the remaining sections of this paper.

3.3. Data Sources and Search Strategy

The identification of initial data sources comes directly from the SLR-SE Guidelines. The electronic sources of relevance to software engineering include:

(online OR on-line OR distance OR MOOC OR
SPOC OR remote* OR e-learn* OR eLearn* OR
internet OR virtual*)
AND
(”software engineering” OR
”requirements engineering” OR ”software design” OR
”software architecture” OR ”software testing” OR
”software project management”)
AND
(education OR training OR course* or teach*)

Figure 1. Search Terms

- IEEExplore
- ACM Digital Library
- Google scholar (scholar.google.com)
- ScienceDirect (www.sciencedirect.com)
- Engineering Village (EI Compendex and Inspec Archive) (https://www.engineeringvillage.com/)

In addition, American Society for Engineering Education (ASEE) PEER (https://peer.asee.org/) was added to those sources from the SLR-SE Guidelines.

In addition to the works identified through these searches, conference proceedings and journals for which a manual inspection of all digitally available publications was completed[1] and resulted in 35 potentially relevant works. The manual inspection was carried out for:

- Conference on Software Engineering Education and Training (CSEE&T)
- International Conference on Software Engineering - Software Engineering Education and Training (SEET) track
- International Conference on Computing Education (ICER) Proceedings
- Koli Calling International Conference on Computing Education Research (Koli)

The manual inspection of the selected sources was carried out first (see §3.7 for inclusion/exclusion criteria). This manual inspection also allowed the researcher to identify keywords which might lead to relevant searched studies. The search terms and boolean operators (AND and OR) used can be found in Figure 1.

[1]A search was also performed on the journal ACM Transactions on Computing Education (formerly Journal on Educational Resources in Computing). No relevant works were found, and it was not included in this list

Table 1. Search Results Statistics

Source	Search Results	Full-text Reviews	Works Used
IEEE Xplore	2681	50	21
ACM Digital Library	286	21	9
Google Scholar	107	10	6
Engineering Village	1948	6	1
ASEE Peer	1951	3	0
ScienceDirect	368	6	0
Search Totals	7237	89	31
Manual Inspection	36	17	10

This general search query had to be customized to each source's search functionality, the details of which can be found in Appendix B. The application of these search terms resulted in 7237 potentially relevant works being identified. See Figure 1 for detailed result statistics. All[2] data captured resulting from the execution of these searches on 2018-03-31 can be found at the web address listed in the appendix.

3.4. Study Relevance Assessment

First, the works identified in the manual inspection were checked against the search results. Of 36 potentially relevant works from the manual inspection, 24 were identified as also having been found by one of the search queries. The identification of 12 potentially relevant works that were not identified by source search underlies the importance of continued manual inspection of likely relevant sources.

Table 2. Manual Inspection Statistics

Source	Potentially Relevant	Duplicated in Search	Full-text Review	Works Used
CSEE&T	22	18	12	7
SEET	7	6	5	3
ICER	1	0	0	0
Koli	6	0	0	0
Totals	36	24	17	10

Next, the remaining works, both manually identified and search results, were analyzed by title and, if necessary, abstract to determine if the work should be included for full-text review (see §3.7 for inclusion/exclusion criteria). This process reduced the number of works to 89[3]. The full text of each of the 89 remaining works was read. Sources were then excluded if they didn't identify its audience nor its specific content

[2]At the time of writing, Wiley Online Library does not have an export feature, so this data could not be exported for capture.

[3]Of the 96 full-text reviews from all sources, 7 were duplicates from another source, leaving 89 full-text reviews to complete.

and if they only described previously or subsequently published results. The final number of sources used for this study was 34[4].

3.5. Study Coding

As part of the SLR-SE Guidelines, a population, intervention, comparison, and outcome (PICO) criteria summary was also generated. The PICO criteria summary is shown in Figure 3 [4].

Table 3. PICO Criteria Summary

Population	Software Engineering
Intervention	Education or Training Course delivered online
Comparison	Audience and content presented
Outcome	Audience and content analysis in online SEET courses

From each work, the following data was extracted:

1. Software engineering learning outcomes addressed by the effort (specifically, SE2014 and GSwE2009 knowledge areas and SWECOM skill sets)

2. Audience classification, as students (further coded as K-12, undergraduate, graduate, post-graduate), industry professionals (further coded as developers, managers, or other), or "none reported."

3. Year of publication of the research

An exemplar of excellent reporting of audience and content area can be found in Edwards' paper from International Conference on Software Engineering (ICSE) 2000. [8] In it, Edwards details not only the course's overall intent and description, but specifically lays out the learning outcomes, making the alignment to SE2014, GSwE2009, and SWECOM very easy. Additionally, Edwards makes clear the various audiences of this graduate course, graduate students with no work experience, to those working full-time. Future such work in online SEET (and, ideally, in SEET as a whole) would derive their content areas directly from the curricula guidelines and/or competency model to make for much easier comparison between studies.

Similarly, an example of the type of paper which was not coded is Soler's A web-based e-learning tool for UML class diagrams [9]. The paper explains the problem, explains the background, details the tool's purpose, and reports experimental findings. However, this is only a tool used within the course. Having an online tool, even if used for an assignment, is not the same as having the whole of the class activities completed online.

[4] 7 coded works were found both in the search results and in the manual search.

3.6. Data Synthesis

Once all studies were properly coded, aggregate data were generated. Studies were grouped by student or industry professional target audience, as well as by the skill set or knowledge area(s) identified in the study. Figures 4, 5, and 6 detail the results of the data synthesis. Note that for each figure, the set/area totals may not equal the sum of the student and industry professional counts due to cases where a study identified both students and industry professionals.

In addition, all coded studies were sorted by year of publication. That data was extracted and plotted in Figure 2.

Lastly, counts of studies reporting students and industry professionals were compared. How many more papers were focused on students rather than industry professionals compared to the total for the content area was calculated.

Table 4. SWECOM Coding

Skill Set	Subject: Students	Subject: Ind. Prof.	Set Total
Requirements	8	4	11
Design	12	3	14
Construction	9	4	11
Testing	5	2	6
Sustainment	4	1	4
Process and Lifecycle	9	5	13
System Engineering	2	0	2
Quality	3	3	5
Security	1	1	1
Safety	0	0	0
Configuration Mgmt	1	0	2
Measurement	2	2	3
HCI	1	0	1

3.7. Inclusion and Exclusion Criteria

During the manual inspection, only inclusion criteria were used. If the scanned title (and abstract preview when available) indicated that the paper may have been relevant, it was included for further review of the abstract and keywords.

Next, the title and abstracts were reviewed to determine if the study would be included in full text review. If the title and/or abstract indicated that the work could be relevant, it was included. This likely

Table 5. GSwE2009 Coding

Knowledge Area	Subject: Students	Subject: Ind. Prof.	Area Total
Ethics	1	0	1
System Engineering	3	0	3
Requirements Engineering	8	4	11
Software Design	13	3	15
Construction	9	4	11
Testing	8	5	11
Maintenance	5	2	5
Configuration Mgmt	1	0	2
Management	7	5	11
Process	7	3	9
Quality	5	2	7

Table 6. SE2014 Coding

Knowledge Area	Subject: Students	Subject: Ind. Prof.	Area Total
Professional Practice	3	1	4
Modeling and Analysis	7	0	8
Req. Analysis and Spec.	8	4	11
Software Design	10	4	13
HCI	1	0	1
V&V	7	5	10
Software Process	12	6	16
Quality	5	2	7
Security	1	1	1

lead to a higher number of false positives (i.e. studies included for full-text analysis which were not relevant), in favor of reducing or eliminating false negatives (i.e. mistakenly excluded a relevant study).

Finally, each included study received a full-text analysis. The study was then excluded if it met any of the following criteria:

- Study did not address software engineering content
- Study did not detail an online delivery of content
- Study detailed a tool or process, rather than reporting of a delivery of content
- Study publication did not have enough detail to determine either audience or content area (e.g. panel discussion summaries or keynote abstract)

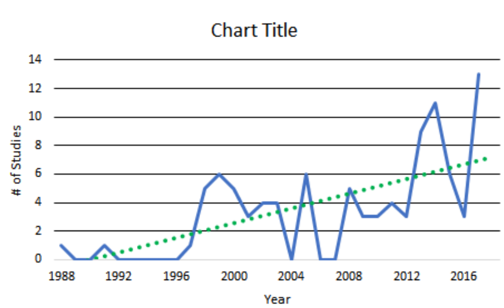

Figure 2. Study Count 1988-2017

4. Results

The result of this process was the selection of 34 studies that clearly identified an audience being delivered software engineering content in an online course, content areas which could be coded into three primary guides/standards: SWECOM, GSwE2009, and SE2014.

4.1. R1: What content is being taught by online SEET courses?

Across all three standards, software design and software process are the most commonly reported course content areas being taught, followed by common life cycle stage education in requirements, construction and testing. The least taught topics are safety, security, ethics, HCI, configuration management, and systems engineering. In the case of every skill set and knowledge area, each has been found to have been taught and reported in at least one study, except for one, safety.

4.2. R2: Who is the audience of online SEET courses?

Thirty studies identified students as a target audience, while only nine identified industry professionals as the target audience. Further, only three studies identified industry professional as a target without also identifying target student populations. Post-graduate students were targeted in only four studies, with only one such study having targeting post-graduate students exclusively. This was similar to studies identifying K-12 students, with only three studies and only one targeting K-12 exclusively.

4.3. R3: How the rate of research of online SEET courses changed over time?

$$y = 0.2581x - 0.8$$

Figure 3. Year over year with trendline

Reviewing the papers which underwent full text analysis by the year of their publication, depicted in Figure 2, an upward trend of approximately .25 can be seen. The trendline equation can be found in Figure 3. Closer inspection also shows an understandable dip shortly after 2001-2002: after the dot-com bubble crash, it would make sense to see a decline in the research, as offering also likely declined. Shortly thereafter, research bounces back, which may indicate a similar bounce back for online courses in general.

5. Discussion

The sheer number of search results made for a time-intensive inclusion process. This was caused primarily by the search terms, which were determined during the manual inspection process. The researcher identified a variety of terms, any of which could have been used to refer to the subject of the paper: online SEET. The variety of synonyms of the word "online" that were found to have been used in potentially relevant papers during the manual inspection necessitated their inclusion. Similarly, the content of a software engineering course could include the description of any of the content areas within the field. This could have meant including each of the skill areas as a search term, for example. Thankfully, only a few such alternatives were found during manual inspection. Many systematic reviews are aided by investigating a niche area with specific (and limited) terminology. In contrast, this paper sought a comprehensive collection of research in a field which is still new and whose terms are not yet universally agreed upon. Once this field expands and settles, the ability to reduce search terms may make future such studies much easier.

Consider the list of skills sets and knowledge areas having the least reported studies identified from Section 4.1. The lack of reporting for instruction in safety, ethics, and, especially, security is concerning. Cyber security continues to be in the news on a regular basis, and the news is not typically good. The lack of research into these areas could indicate a lack of offerings. The topic of safety is the only topic from any of the three standards which was not reported at all in our search. Searches outside of the systematic review completed in Google Scholar ("online software safety course," for example) resulted in zero results. More information is needed to determine the cause for this omission from the literature. Possibilities include missing search terms, confusion about the topic, and difficulty in instructional design for the topic. In addition, if we restrict the role of safety to safety-critical systems, the topic may yet be too niche to support an online course offering, given the higher costs associated with developing materials for them. Systems engineering also saw low reporting numbers. This could likely be due to the search terms, given that "software engineering" rather than "systems engineering" was searched. This would likely be served well by further inspection. With these gaps now identified, researchers have a much clearer picture of what software engineering topics remain uninvestigated, and, potentially, not offered at all in the online landscape.

The audience for the reporting content delivery is skewed quite heavily towards students enrolled in traditional (classroom-based) degree programs, specifically undergraduate and then graduate students. This is likely to be expected: most researchers are either faculty or graduate students at institutions of higher education. Even as software design has the highest number of reporting studies, the numbers favor students compared to industry professionals. In many cases, the benefits of software design do not become apparent until the learner is forced to continue developing code over a longer period of time. This allows for the necessity of refactoring and extension to become necessary. Learning software design often focuses on these sorts of benefits, often lost when the single-use assignments are completed, never to be revisited, avoiding the pitfalls of low-quality (even if functional) code. Despite this, more emphasis is seen in providing this content to students than to industry professionals, when it is possible that industry professionals would see greater gains, grounded in their experience.

A non-trivial number of studies were excluded from our study for the same reason: describing a tool rather than the results from online content delivery. Much like general SEET research, these tool papers are isolated from each other. A survey of online SEET tools may be of interest to practitioners. Ten such studies were identified by this systematic review, with more tools likely to be identified in a search more specifically targeted at finding tool papers.

The validity and efficiency of the search terms was also an area of interest. The wide-range of potentially relevant keywords led to a high number of search results, as high as 9300 potentially relevant works in an early version of the searches to as many as 163,000 raw search results for some sources. Now that the relevant works have been identified, a review of the search terms used could potentially reduce the number of irrelevant search results through a revision of the search terms used. This could then serve as a basis for a more standardized set of terminology for future research and future systematic literature reviews.

Using the highest search results' source, IEEE Xplore as an example, removing these two search terms resulted in approximately 10% fewer search results. Table 7 shows the results of this analysis. In general, each search term resulted in search results. "eLearn" and "software testing" resulted in zero coded works. These could be candidates for removal from the search terms. However, this reduction in the search terms may lead to relevant studies being excluded if such studies are published after the time of this writing.

6. Threats to validity

The primary threat to validity is the exclusion, inclusion, and coding of studies by a single researcher. When these tasks are completed by more than one researcher, the results can be protected from false negatives (excluding a study when it should be included or not coding a study as having addressed a skill set or knowledge area when it had) by comparisons between researcher results. Without the parity check, there is a higher chance of human error affecting the results. This possibility is addressed by providing the full parameters of the search terms for each source, capturing the search results and publicizing them, and allowing for easy external verification of the coding by making available the set of references for those papers which were provided a full-text review.

Another threat is the omission of the CiteSeer search source. When the search term was converted into a format which was acceptable to CiteSeer, a search of title and abstracts resulted in 163,756 results. This was not a practical amount of results to parse. Restricting the search to titles resulted in a minimal reduction: 131,074 results. Instead, a manual inspection was completed of many pages of results. No unique studies were found.

7. Conclusions

Software engineering is being taught online. From small, private courses to massively open ones, content is being delivered. And, while research was being completed, there was little visibility into how well software engineering education and training at scale is being researched. The systematic review that has been performed provides this "big picture" visibility. Several topics are well represented, while others need more attention. Future work is necessary by educators and researchers to investigate these essential areas of software engineering, especially security.

Systematic reviews in software engineering education will remain time-consuming while the terminology used by researchers continues to be varied. Efforts that would seek to standardize the language used to refer to online SEET and related concepts may lead to easier research into these topics. Modified guidelines for the performance of systematic literature reviews where software engineering meets education and training would be beneficial.

Gaps in the research have now been established, especially surrounding important and in-demand topics like security. A clear picture has now emerged of the research which has and has not been undertaken in online software engineering education and training, identifying missing areas in topics like software safety and for audiences like K-12 studets and industry professionals who are not also students.

APPENDICES

A. Systematic Review Data

Complete search results exports, full-text review analysis results, and references to all studies which received a full-text review (available as Bibtex) are available at `http://www.cs.umn.edu/~wendt/papers/CSEET_31/data/`

B. Exact Search Terms Per Source

B.1. IEEE

(online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) AND (”software engineering” OR ”requirements engineering” OR ”software design” OR ”software architecture” OR ”software testing” OR ”software project management”) AND education

(online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) AND (”software engineering” OR ”requirements engineering” OR ”software design” OR ”software architecture” OR ”software testing” OR ”software project management”) AND course*

(online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) AND (”software engineering” OR ”requirements engineering” OR ”software design” OR ”software architecture” OR ”software testing” OR ”software project management”) AND training

(online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) AND (”software engineering” OR ”requirements engineering” OR ”software design” OR ”software architecture” OR ”software testing” OR ”software project management”) AND teach*

Query searched abstract, title text, and indexing terms. Query run under the advanced search's Command Search functionality, allowing the above to be directly inserted. Metadata only was selected. Due to the search term limit, the first two parenthesized conditions were paired with each of the search terms in the final condition one at a time (see above for exact

Table 7. Coded Work Count by Search Term

	"software engineering"	"requirements engineering"	"software design"	"software architecture"	"software testing"	"software project managment"	
	34	2	5	2	0	2	
	online	on-line	distance	MOOC	SPOC	remote*	
	21	2	19	7	1	2	
e-learn*	eLearn*	internet	virtual*	education	training	course*	teach*
3	0	6	4	34	12	30	20

search terms). Results were combined in Google Sheets and duplicates identified and removed. This manual process could have resulted in an inadvertent deletion of a relevant work, but care was taken to avoid such mistake (including re-execution of the queries during the process of deciding which items to include for full-text analysis).

B.2. ACM

(acmdlTitle:(online distance MOOC SPOC remote e-learn* eLearn* internet virtual*) OR recordAbstract:(online distance MOOC SPOC remote e-learn* eLearn* internet virtual*) OR keywords.author.keyword:(online distance MOOC SPOC remote e-learn* eLearn* internet virtual*)) AND (acmdlTitle:("software engineering" "requirements engineering" "software design" "software architecture" "software testing" "software project management") OR recordAbstract:("software engineering" "requirements engineering" "software design" "software architecture" "software testing" "software project management") OR keywords.author.keyword:("software engineering" "requirements engineering" "software design" "software architecture" "software testing" "software project management")) AND (acmdlTitle:(education training course* teach*) OR recordAbstract:(education training course* teach*) OR keywords.author.keyword:(education training course* teach*))

Query searched abstract, title text, and author's keywords. Query run under the advanced search's Edit Query functionality, allowing for this query to be directly inserted.

B.3. ScienceDirect

tak(online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) AND tak("software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management") AND tak(education OR training OR course* OR teach*)[Journals(Computer Science,Engineering)]

Query searched abstract, title, and keywords. Query run under the advanced search's Expert search functionality. Searches were restricted to Journals (not Books) in the Computer Science and Engineering areas (in the multiple-selection dropdown).

B.4. Engineering Village

((online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) wn KY AND ("software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management") wn KY AND (education OR training OR course* OR teach*) wn KY NOT "Institute of Electrical and Electronics Engineers Inc." wn PN NOT "Association for Computing Machinery" wn PN NOT "Association for Computing Machinery, Inc" wn PN NOT IEEE wn PN NOT ACM wn PN NOT "American Society for Engineering Education" wn PN NOT ASEE wn PN)

Query searched abstract, title, and subject. Query run under search's Expert functionality. Certain publishers were excluded to reduce the number of records to be read, from 3980 to 1948. Publishers with their own search engines (ACM, IEEE, and ASEE) were excluded using query syntax.

B.5. ASEE Peer

(online OR on-line OR distance OR MOOC OR SPOC OR remote* OR e-learn* OR eLearn* OR internet OR virtual*) AND ("software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management") AND (education OR training OR course* OR teach*)

Query searched without restriction. Query run under advanced search. Query was placed in the "Look For" field unedited. Selecting any "Only In" other than "Document Content" in any combination resulted in one (irrelevant) work. The difference between restricting search to only document content was 9 (1974 with document content only restriction vs. 1983 without.). At the time of this search, only the first 1000 results could be exported. Each item was reviewed on the website.

B.6. Google Scholar

allintitle: education online OR distance OR MOOC OR virtual OR remote "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: course online OR distance OR MOOC "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: training online OR distance OR MOOC "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: teach online OR distance OR MOOC "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: education SPOC OR remote OR internet OR virtual "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: course SPOC OR remote OR internet OR virtual "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: training SPOC OR remote OR internet OR virtual "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management" allintitle: teach SPOC OR remote OR internet OR virtual "software engineering" OR "requirements engineering" OR "software design" OR "software architecture" OR "software testing" OR "software project management"

Queries searched on title only (only restriction available). Queries run under standard search.

References

[1] E. Roberts, T. Camp, D. Culler, C. Isbell, and J. Tims, "Rising CS Enrollments: Meeting the Challenges," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, (New York, NY, USA), pp. 539–540, ACM, 2018.

[2] Committee on the Growth of Computer Science Undergraduate Enrollments, Board on Higher Education and Workforce, Policy and Global Affairs, Computer Science and Telecommunications Board, Division on Engineering and Physical Sciences, and National Academies of Sciences, Engineering, and Medicine, *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments*. Washington, D.C.: National Academies Press, Mar. 2018.

[3] J. Friedman, "4 Expectations for Online Education in 2018," *U.S. News & World Report*, Jan. 2018.

[4] B. Kitchenham and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007.

[5] IEEE Computer Society, "Software Engineering Competency Model," p. 168, 2014.

[6] ACM/IEEE-CS Joint Task Force on Computing Curricula, "Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," 2014.

[7] Integrated Software & Systems Engineering Curriculum (iSSEc) Project, "Graduate Software Engineering 2009: Curriculum Guidelines for Graduate Degree Programs in Software Engineering," Sept. 2009.

[8] S. Edwards, "Can quality graduate software engineering courses really be delivered asynchronously on-line?," in *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, pp. 676–679, June 2000.

[9] J. Soler, I. Boada, F. Prados, J. Poch, and R. Fabregat, "A web-based e-learning tool for UML class diagrams," in *IEEE EDUCON 2010 Conference*, pp. 973–979, Apr. 2010.

[10] K. A. Gary, S. Sohoni, and T. Lindquist, "It's Not What You Think: Lessons Learned Developing an Online Software Engineering Program," in *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*, pp. 236–240, Nov. 2017.

[11] L. J. White and J. Coffey, "The design and implementation of an innovative online program for a master of science degree in Computer Science #x2014; Software Engineering specialization," in *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE T)*, pp. 257–265, May 2011.

[12] J. Armarego, L. Fowler, and G. G. Roy, "Constructing engineering knowledge: development of an online learning environment," in *14th Conference on Software Engineering Education and Training, 2001. Proceedings*, pp. 258–267, 2001.

[13] G. W. Hislop, "An online system analysis course," in *Proceedings 12th Conference on Software Engineering Education and Training (Cat. No.PR00131)*, pp. 138–144, Mar. 1999.

[14] H. J. C. Ellis, "Software engineering at a distance," in *Proceedings 11th Conference on Software Engineering Education*, pp. 23–34, Feb. 1998.

[15] G. W. Hislop, "Teaching via asynchronous learning networks," in *Proceedings 11th Conference on Software Engineering Education*, pp. 16–22, Feb. 1998.

[16] F. Hermans and E. Aivaloglou, "Teaching Software Engineering Principles to K-12 Students: A MOOC on Scratch," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, pp. 13–22, May 2017.

[17] W. Billingsley and J. R. H. Steel, "Towards a Supercollaborative Software Engineering MOOC," in *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, (New York, NY, USA), pp. 283–286, ACM, 2014.

[18] J. Hudepohl, A. Dubey, S. Moisy, J. Thompson, and H.-M. Niederer, "Deploying an Online Software Engineering Education Program in a Globally Distributed Organization," in *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, (New York, NY, USA), pp. 301–310, ACM, 2014.

[19] D. Carrington, S.-K. Kim, and P. Strooper, "An Experience Report on Using Collaboration Technologies for Distance and On-campus Learning," in *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*, ACE '10, (Darlinghurst, Australia, Australia), pp. 45–52, Australian Computer Society, Inc., 2010.

[20] C. M. Calongne, "Promoting Team Interaction in the Online Classroom," *J. Comput. Sci. Coll.*, vol. 18, pp. 218–227, Oct. 2002.

[21] K. Marasovi and M. Lutz, "Remote Development and Distance Delivery of Innovative Courses: Challenges and Opportunities," in *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ICSE '15, (Piscataway, NJ, USA), pp. 299–302, IEEE Press, 2015.

[22] C. Murphy, D. Phung, and G. Kaiser, "A Distance Learning Approach to Teaching eXtreme Programming," in *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '08, (New York, NY, USA), pp. 199–203, ACM, 2008.

[23] J. Cabot, F. Durn, N. Moreno, A. Vallecillo, and J. R. Romero, "From Programming to Modeling: Our Experience with a Distributed Software Engineering Course," in *Proceedings of the 30th International Conference on Software Engineering*, ICSE '08, (New York, NY, USA), pp. 749–758, ACM, 2008.

[24] D. C. Schmidt and Z. McCormick, "Producing and Delivering a Coursera MOOC on Pattern-oriented Software Architecture for Concurrent and Networked Software," in *Proceedings of the 2013 Companion Publication for Conference on Systems, Programming, & Applications: Software for Humanity*, SPLASH '13, (New York, NY, USA), pp. 167–176, ACM, 2013.

[25] M. A. Ardis and P. B. Henderson, "Software Engineering Education (SEEd): Is Software Engineering Ready for MOOCs?," *SIGSOFT Softw. Eng. Notes*, vol. 37, pp. 14–14, Sept. 2012.

[26] M. Wermelinger, J. G. Hall, L. Rapanotti, L. Barroca, M. Ramage, and A. Bandara, "Teaching Software Systems Thinking at the Open University," in *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ICSE '15, (Piscataway, NJ, USA), pp. 307–310, IEEE Press, 2015.

[27] K. D. Wendt, K. Reily, and M. P. E. Heimdahl, "First Steps towards Exporting Education: Software Engineering Education Delivered Online to Professionals," in *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, pp. 241–245, Apr. 2016.

[28] H. J. C. Ellis, "Autonomous Learning in Online and Traditional Versions of a Software Engineering Course," in *18th Conference on Software Engineering Education Training (CSEET'05)*, pp. 69–76, Apr. 2005.

[29] C. Theisen, L. Williams, K. Oliver, and E. Murphy-Hill, "Software Security Education at Scale," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pp. 346–355, May 2016.

[30] L. Shafer, "Distance learning need not sacrifice high-quality education," in *31st Annual Frontiers in Education Conference. Impact on Engineering and Science Education. Conference Proceedings (Cat. No.01CH37193)*, vol. 1, pp. T3F–2–8 vol.1, 2001.

[31] J. Paynter and D. Sheridan, "Using the Internet to teach software engineering," in *Proceedings. 1998 International Conference Software Engineering: Education and Practice (Cat. No.98EX220)*, pp. 312–321, Jan. 1998.

[32] A. G. d. O. Fassbinder, M. Fassbinder, E. F. Barbosa, and G. D. Magoulas, "Massive open online courses in software engineering education," in *2017 IEEE Frontiers in Education Conference (FIE)*, pp. 1–9, Oct. 2017.

[33] N. Kilicay-Ergin and P. A. Laplante, "An Online Graduate Requirements Engineering Course," *IEEE Transactions on Education*, vol. 56, pp. 208–216, May 2013.

[34] J. S. Collofello, "Development of a web-based introduction to software engineering short course for working professionals," in *32nd Annual Frontiers in Education*, vol. 1, pp. T4E–8–T4E–11 vol.1, 2002.

[35] C. Lpez, J. M. Alonso, R. Marticorena, and J. M. Maudes, "Design of e-activities for the learning of code refactoring tasks," in *2014 International Symposium on Computers in Education (SIIE)*, pp. 35–40, Nov. 2014.

[36] S. Rehberger, T. Frank, and B. Vogel-Heuser, "Benefit of e-learning teaching C-programming and software engineering in a very large mechanical engineering beginners class," in *2013 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1055–1061, Mar. 2013.

[37] K. Bothe, K. Schutzler, Z. Budimac, and K. Zdravkova, "Collaborative development of a joint Web-based software engineering course across countries," in *Proceedings Frontiers in Education 35th Annual Conference*, pp. S3H–7, Oct. 2005.

[38] I. M. S. Gimenes, L. Barroca, and E. F. Barbosa, "International workshop on distance learning support for postgraduate programs in software engineering (e-gradSE)," in *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE T)*, pp. 517–519, May 2011.

[39] K. Zdravkova, K. Bothe, and Z. Budimac, "SETT-Net: a network for software engineering training and teaching," in *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.*, pp. 281–286, June 2003.

[40] L. Douglas, "Learning object-oriented software design at a distance," in *Frontiers in Education Conference, 1999. FIE '99. 29th Annual*, vol. 2, pp. 12C2/24–12C2/27 vol.2, Nov. 1999.

[41] S. Efremidis, S. Retalis, N. Papaspyrou, and E. Skordalakis, "Teaching Software Engineering through the Net," *WIT Transactions on Information and Communication Technologies*, p. 10, 1997.

[42] M. Jacobsen, R. Kremer, and M. Shaw, "An Examination of Distance Education Learning Environments in Two Software Engineering Graduate Courses," May 1999.