World Academy of Science, Engineering and Technology
International Journal of Educational and Pedagogical Sciences
Vol:6, No:11, 2012

A Systematic Mapping Study on Software Engineering Education

Bushra Malik, and Saad Zafar

***Abstract*—**Inadequate curriculum for software engineering is considered to be one of the most common software risks. A number of solutions, on improving Software Engineering Education (SEE) have been reported in literature but there is a need to collectively present these solutions at one place. We have performed a mapping study to present a broad view of literature; published on improving the current state of SEE. Our aim is to give academicians, practitioners and researchers an international view of the current state of SEE. Our study has identified 70 primary studies that met our selection criteria, which we further classified and categorized in a well-defined Software Engineering educational framework. We found that the most researched category within the SE educational framework is *Innovative Teaching Methods* whereas the least amount of research was found in *Student Learning and Assessment* category. Our future work is to conduct a *Systematic Literature Review* on SEE.

***Keywords*—**Mapping Study, Software Engineering, Software Engineering Education, Literature Survey.

I. Introduction

SOFTWARE Engineering (SE) is an evolving field therefore; SE education must prepare its students for constant learning, to enable them to move beyond today's technology and to meet the challenges of the future [1]. To this end, there is a general consensus that the Software Engineering Education (SEE) in 21st century needs to move forward from lecture format to other varieties of learning and teaching approaches [1].

Inadequate curriculum for software engineering is considered to be one of the most common software risks [2]. To engineer the complex and critical software systems, the industry requires those software engineers who possess appropriate skills, knowledge, and expertise [3]. Although, the increasing demand of software professionals in government and business sector has increased pressure on academic institutes to produce greater number of competent software developers [4], still there is a considerable gap between the topics taught to students in university courses and the skills and practical knowledge required by the industry [5]. Due to this gap, companies have to provide an extra training to fresh graduates before assigning actual jobs to them [6]. Therefore, the aim of SEE should be to prepare its students, i.e. future software engineers, for different roles, and should instill a stronger engineering attitude in them [7].

B. Malik and S. Zafar are with the Faculty of Computing, Riphah International University, Islamabad, Pakistan (phone: 92-51-5469643; fax: 92-515469636; e-mail: bushra.malik@riphah.edu.pk; saad.zafar@riu.edu.pk).

The aim of this study is to give an international view of the current state of SEE, to academicians, practitioners and researchers, by identifying the latest advancements taken place to improve the state of SEE. For this purpose, we have conducted a mapping study (defined in Section II) on SEE. The objective of conducting this study was also to check the feasibility of conducting SLR [8] on SEE, which is our current ongoing project.

Our mapping study has addressed the following three research questions. *RQ1*: What has been researched with regard to the education of software engineering? *RQ2*: Which Conferences/Journals included the above research? *RQ3*: Which country is leading SE educational Research?

We have mapped our primary studies on a well-defined SE educational framework [4]. Our results show that *Innovative Teaching Methods* (ITM) are being utilized more to improve the state of SE education [4]. But the category; *Student Learning and Assessment* (SL&A) was researched, the least.

The remainder of the paper is organized in the following way: Background of our study is explained in Section II. Section III describes the process of how we conducted our mapping study, including the protocol of our research, selection procedure of primary studies and analysis, classification of these primary studies. Discussion on our study results is given in given Section IV. Study Limitation and Threats to Validity are identified in Section V. Conclusion and Future Work are given in Section VI.

II. Background

Either theoretically or practically, teaching Software Engineering is not an easy job [9]. The research on SEE has reported a number of SE educational problems, such as: difficulties faced in learning distributed software development [10], [11], ignoring work cultural issues in SE courses [12], limited practice of using peer reviews in software design projects [13], ignoring the issue of formal methods in SE teaching [14], highlighting the fact that only classroom learning model is an ineffective approach towards SE teaching [15]. Thus, SEE needs to be more realistic and learner-centered [16].

Nevertheless, research has also showed that significant amount of work has been done to improve SEE. Some of those innovations are: developing a Master’s program on Software Management including the aspect of business of software [17], adoption of coaching as a major teaching model [18], improving advanced software development projects with the help of LEJOS and Mindstorms [19], enhancing the software testing teaching by using Metaphoric Testing (MT); an end-user methodology [20], interaction of students with real clients in software practicum course [21], proposal of a framework for Contextualised Software Engineering Education (CSE2) [22].

To identify the extent of these evidences, we have conducted a mapping study on SEE, as Mapping Study outlines the extent of the research area based on the specific research question(s) [23]. It is also referred to as a Scoping Study [8] and is a predecessor of a Systematic Literature Review (SLR) [23]. Mapping study represents the evidence at the higher level of granularity to identify evidence clusters and evidence deserts [8]. The evidence cluster sets paths for future SLRs whereas evidence deserts identify the need for more research on that particular area [8]. Therefore, we have conducted this mapping study to check if there exists an evidence cluster for SLR on SEE.

III. A Systematic Mapping Study Process

We followed the mapping study process using the template protocol given on the Evidence-Based Software Engineering (EBSE) website [24] and from Kitchenham's guidelines [8]. Based on these sources, our mapping study process contained the following three stages: (1) Defining Scope, Search Strategy and Selection Criteria; (2) Selecting Primary Studies; and (3) Analysis, Classification and Map building.

A. Step1: Defining Scope, Search Strategy and Selection Criteria

To obtain success in a systematic mapping study, we need to plan it properly [25]. Therefore, in this section, we have defined the scope, search strategy and the selection criteria of our study. The scope of our study is to investigate the latest techniques, technologies and methodologies proposed/adopted or implemented in Software Engineering Education. Keeping in mind the scope of our study, we went through an iterative process before finalizing our search strategy and selection criteria. Based on the suggestions of our experts (see Acknowledgment Section) and on the experience we gained in our pilot study, we developed the following search strategy for our mapping study. One anonymous expert suggested not to use complex Boolean terms and proposed to use the following search string in our study:

'Software Engineering Education'

Because according to her, this search string would itself yield a large number of false positives, which we found true after executing this search string. She also advised us to keep our search process in a manageable amount of time. Therefore, initially, we searched the relevant literature on only two sources; which were ACM digital Library (DL) and IEEE Xplore. Expectedly, we identified a substantial amount of primary studies from these two sources and thus, did not proceed to other sources (See Table I). To collect the evidence relevant to our research questions, we used the following *inclusion criteria.* (1) The research papers related to improve SE education published between January 2010 and October 2010. (2) The paper's abstract must explicitly mention research on software engineering education, and the Title, Abstract, Introduction or Keywords must contain one of the following words: Software Engineering, Software Engineers, or Software Engineering Education. Our *exclusion criteria* excluded those papers that mentioned SE education but their context was contrary to the scope of our study.

During our pilot study, we found that a significant amount of research has been conducted on SEE, that is why, in this study, we have included only those research papers, which were published in the time span of January 2010- October 2010, in order to give the latest innovations in SE educational research.

B. Step 2: Selecting Primary Studies

We searched our databases; IEEE Xplore and ACM Digital Library (DL) with our search string;"Software Engineering Education" and found a total of 1,589 papers (See Table I). While searching IEEE Xplore, we further refined our search by setting publication year as 2010 and manually selecting Journal or conference proceedings option. While searching ACM DL, we made three automated refinements by limiting our search to:

The ACM Guide to Computing Literature

Publication year as 2010

Journal or proceedings publications

In addition to the above refinements, we also manually checked the dates of all relevant primary studies (of both IEEE Xplore and ACM DL) to include only those studies which were published till 31st October, 2010 (as we started our search process on 1st November and it took a couple of days to complete it). On the basis of our inclusion/exclusion criteria, we read the title, abstract and set of keywords of all 1,589 papers and initially selected 77 primary studies.

TABLE I
Number of Primary Studies, Source-Wise

Databases	Total No. of Primary Studies	No. of selected Primary Studies	No. of Relevant Primary Studies	Percentage of Relevant Primary Studies
IEEE Xplore	751	56	51	6.79 %
ACM DL	838	21	19	2.27 %
Total	1,589	77	70	4.40 %

In order to select relevant primary studies, we critically read the abstracts of all 77 primary studies in depth, where necessary or desired, sections of introduction and conclusion were also read. As a result of this step, we extracted the final 70 relevant primary studies for our mapping study (See Table I). The rest of 7 studies were rejected as those studies did not report on how to improve SEE.

Initially, we found more papers in ACM DL (838/1589 or 52.7%) but eventually IEEE Xplore yielded the most relevant (51/70 or 72.8%) papers. (See Table I). We used EndNote [26]; a bibliographic management software, to effectively manage the references of our primary studies. We developed our data extraction form in a word document.

A. Step 3: Analysis, Classification and Map Building

The purpose of this step is to build systematic maps; maps which will depict the evidence clusters and evidence deserts. But to accomplish this, we first need to analyze and classify our primary studies. Thus, further in this section, we will answer our three research questions:

RQ1: What has been researched with regard to the education of software engineering?

This question aims to identify the recent trends in research to improve the SE education. To this end, first we analyzed all our relevant primary studies, so that we could develop a set of categories for our primary studies. The definition of categories, also called as categorization, begins with abstract reading, and sections of introduction and conclusion can also be read, where necessary or desired [25]. After that, Keywords and perception related to the contribution of each primary study is identified [27]. A set of keywords from different papers were combined together to develop a high level understanding about the nature and contribution of the research. When a final set of keywords were chosen, they can be clustered and used to form the categories for the map.

During category identification, we had two choices; either to find a relevant Software Engineering Educational framework from the literature or to make one from our own in accordance with our primary studies. We found only one Framework intended for SE educational research; given in the book "Software Engineering: Effective Teaching and Learning Approaches and Practices" written by Ellis *et al* [4]. This SE educational framework [4] contains six categories. They are:

Innovative Teaching Methods: Research based on the varieties of teaching methods; which are innovative in their own way. The goal behind innovation should be to improve student's skill set and to enhance their abstract thinking, awareness and learning.

Curriculum and Education Management (C&EM): Research based on how a new or revised curriculum has been developed/implemented according to the current needs of SE field. It also includes research based on how other educational management issues have been addressed or improved.

Educational Technology (ET): Research based on how e-learning technologies are improving the education of SE.

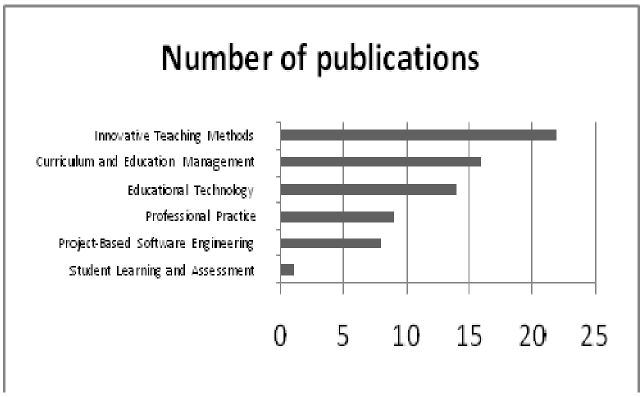

Fig. 1 Software Engineering Educational Map 1: Primary Studies categorized in Ellis et al Framework

Professional Practice (PP): Research based on how to polish the professional and ethical values of students via SE courses/curriculum.

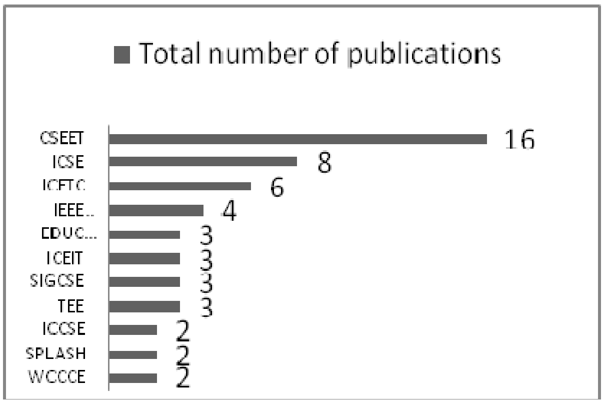

Fig. 2 Software Engineering Educational Map 2: Top Publication Channels

Project-Based Software Engineering (PB-SE): Research based on how to refine student's learning by giving them the opportunity of practically applying their knowledge in project-based software engineering courses; including capstone projects.

Student Learning and Assessment: Research based on how learning and teaching theories can improve student's learning.

Expectedly this framework [4] covered all the categories of our primary studies, adequately. Thus, we have categorized our primary studies according to the framework provided by Ellis et al, as shown in Fig. 1. This framework has also been used in some other contributions [28]-[31] SE educational research.

*RQ2***:** Which Conferences/Journals are leading the above research?

This question aims towards identifying those conferences and journals in which our primary studies have been published. After collecting the conference and journal information of all primary studies, we found that our primary studies appeared in 28 conference proceedings and only in 1 journal. 'IEEE Transactions on Education' was the only journal, in which we found 4 primary studies. Our rest of 66 primary studies appeared in 28 different conference proceedings. Fig. 2 shows our 11 top publication channels: CSEET, ICSE, ICETC, IEEE Trans. Educ, EDUCON, ICEIT, SIGCSE, TEE, ICCSE, SPLASH and WCCCE. The rest of 18 conferences (not shown in Fig. 2) had one publication, each. The names of those conferences are: ACHI, ASWEC, CIT, ICBECS, ICCRD, ICCSNA, ICETC, ICEE, ICENT, ICER, ICIS, ITHET, ITiCSE, ITNG, OOPSLA, SEDM, SIGITE, and SoutheastCon.

RQ3: Which countries are leading the SE educational Research?

This question aims towards identifying those countries which are leading the SE educational research. For this purpose, we categorized our primary studies by author's country of institution, where we found a total of 23 participating countries, in total. We found that many papers had multiple authors, which is a norm in most of fields, including software engineering. But the interesting thing was that many times those multiple authors belonged from different institutes, sometimes from different countries and even sometimes from different continents. To effectively synthesize the variant nature of these results, we distributed the contribution of countries as an Individual or Collaborative contribution, as shown in Fig. 3.

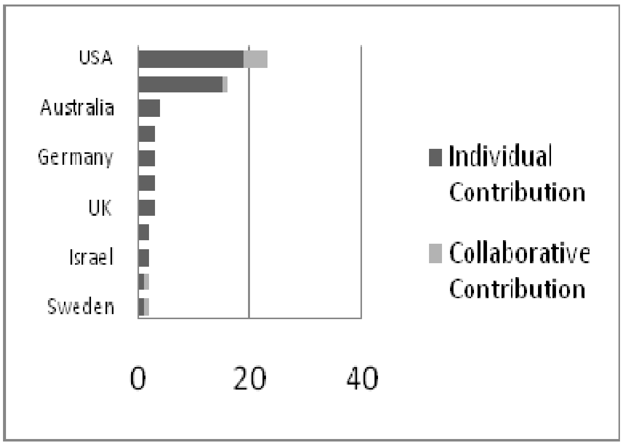

Fig. 3 Software Engineering Educational Map 3: Top countries leading SE Educational Research

Fig. 3 show that USA and China are leading the SE educational research. Australia, Canada, Germany, Malaysia and UK have also contributed significantly. The countries with minimum but more than one publication are: Brazil, Israel, Pakistan and Sweden. The 12 countries, having a single publication, not shown in Fig. 3, contributed individually or collaboratively. The names of countries with a single-individual publication are: Iran, Ireland, Japan, Korea, Mexico, Namibia, Serbia, Spain and Switzerland and the names of the countries with a single-collaborative publication are: Croatia, India and Turkey.

IV. Discussion

Our discussion is split into three sections: the first section of our discussion will relate the 4-tier taxonomy of software exporting nations [32] with the results, we obtained in RQ3. The second section shows the relationship between our primary studies and reference curriculums [1], [33], [34]. The third section discusses the results we obtained after answering all our research questions.

A. Comparison of data identified in RQ3 with Carmel's 4-tier taxonomy [32]

IT industry is the main consumer of software engineering graduates; thus, it must have a positive attitude towards education and should cooperate with academia [35]. On the other hand, Higher Education must involve IT industry while the curriculum development and policy making [35]. It is because active participation of industry plays an important role in the success of a software engineering program [1]. Therefore, in this section, we have compared the 4-tier taxonomy of the world's software exporting nations [32], shown in Table II, with the data we identified in RQ3. Carmel [32] published this 4-tier taxonomy in 2003, but to the best of our knowledge, no new or updated taxonomy on the software exporting nations is available in literature. Carmel's taxonomy has also been referred in [36], [37].

Carmel has stated in [32] that he did not include some of the nation's names due to lack of data availability. As a consequence, out of our 23 participating countries, 5 countries have not been mentioned by Carmel [32].

TABLE II
THE 4-TIER TAXONOMY OF THE WORLD'S SOFTWARE EXPORTING NATIONS, REPRODUCED FROM [32]

	Label	Nations
Tier 1	Major software exporting nations	Mostly OECD nations such as: USA, Canada, UK, Germany, France, Belgium, Netherlands, Sweden, Finland, Japan, Switzerland, Australia. Includes entrants from the 1990s: Ireland, Israel and India.
Tier 2	Transition Software exporting nations	Only Russia and China.
Tier 3	Emerging Software exporting nations	Brazil, Costa Rica, Mexico, Philippines, Malaysia, Sri Lanka, Korea, Pakistan, Romania, Bulgaria, Ukraine, Poland, Czech Republic, Hungary ,others [a]

Tier 4	Infant stage software exporting nations	Cuba, El Salvador, Jordan, Egypt, Bangladesh, Vietnam, Indonesia, Bangladesh, Iran, others. [b]
Non-Competing	Non-Competing	Most of the (smaller, least developed) countries of the world. [c]

a-Several other countries are likely to be in Tier 3: Estonia, Latvia, Lithuania, Chile, Argentina, Thailand, and South Africa.
b-Another 10-20 nations are likely to be in this tier though data are not available.
c- Non-Competing nations in software exports include many of the least developed nations. These include most African nations (e.g., Gambia, Nigeria, and Mozambique) and many of the least develop nations in America (e.g., Bolivia, Paraguay) and Asia (e.g., Syria, Afghanistan, Laos).

Those countries are: Spain, Namibia, Serbia, Slovak Republic and Turkey. As all of these mentioned countries participated in our study, with a single publication, each, hence, we considered it as a minor deviation, and continued our discussion based on Carmel's taxonomy. We compared our rest of the 18 participating countries, as specified in Section III, with the countries mentioned in Table II. Separately, from each tier, we counted the number of countries which participated in our SE education research and showed its result in Table III, given below.

TABLE III
NUMBER OF COUNTRIES PARTICIPATED IN SE EDUCATION RESEARCH, TIER-WISE

Tier #	Number of publications
Tier 1	44
Tier 2	16
Tier 3	9
Tier 4	1
Non-Competing	Nil

Table III shows that the countries of high level tiers are not only the leading Software Exporting Nations [32] but are also the nations which are leading in the SE educational Research. Countries like USA, Australia, Canada, etc. although being in Tier 1 are still improving their SE education. As the stability of the 'Software Exporting Nations' comes down, so does their contribution, regarding SE educational research is reduced (See Table III).

No doubt there are a couple of nations being in tier 1 or tier 2, as shown in Table II; like India, Belgium, Russia etc., but they have a minimum or no publication at all, as discussed previously, in section III. That is why, we did not claim that there lies a co-relation between Carmel's 4-tier taxonomy [32], shown in Table II and in our 23 participating countries. But the reason why we did this comparison, was to motivate the educationists, researchers and practitioners of Emerging Software Exporting nations (tier 3), Infant Stage Software Exporting Nations (Tier 4) and Non-Competing Software Exporting Nations to start their efforts on improving their current state of SE education in order to become a 'Major Software Exporting Nations'; like USA, Australia etc.

A. Relationship between Primary Studies and Reference Curriculums

As we collected literature on software engineering education, so we found it relevant to draw a relationship between our primary studies and SE 2004 (Software Engineering Curriculum Guidelines for Undergraduate, 2004) [1], GSwERC (Graduate Software Engineering Reference Curriculum) [33] and SWEBOK (Software Engineering Body of Knowledge) [34], in order to find relevance of our primary studies with both education and practice. Thus, for education, we used SE 2004 [1] and GSwERC [33] and for practice, we used SWEBOK [34], as our baselines. SE 2004 [1] and GSwERC [33] are curriculum guidelines for undergraduate and graduate degree programs, respectively, whereas SWEBOK [34] identifies a set of body of knowledge for a software engineer professional who has at least five years of experience in the SE field. The same kind of mapping of primary studies on SE2004 [1] and SWEBOK [34] can also be found in the studies of Kitchenham *et al* [38] and Silva *et al* [39].

Before mapping our primary studies on reference curriculums [1], [33], [34], we first distributed our primary studies with respect to the degree programs, as shown in Fig. 4.

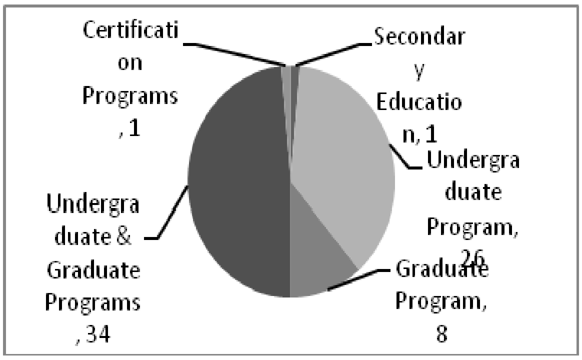

Fig. 4 Distribution of Primary Studies; Degree Program wise

Sometimes, it was explicitly mentioned in the study that the study is meant for undergraduate or graduate or for both - levels. For example, in study like [14] it was stated that it is for undergraduate program or like the study [40] declared that it was for both undergraduate and graduate programs. Therefore, we distributed our primary studies likewise. Otherwise, if the study did not mention for which degree it was meant for, we classified those studies under undergraduate, graduate or to both levels, by understanding the context of paper. It is shown in Fig. 4 that those primary studies which aimed towards both undergraduate and graduate curricula, together, have been found most in this study.

While mapping primary studies to reference curriculums [1], [33], [34], we found it difficult to map some of our primary studies on SE2004 [1], GSwERC [33] and SWEBOK

[34]. Therefore, we consulted Fabio Q.B. da Silva; the author of "Six years of systematic literature reviews in software engineering: An updated tertiary study" [39] in order to take help in mapping those studies. In a direct correspondence with Silva [39], we came to know that sometimes it might be a possibility that some research topics or areas cannot be mapped to SE2004 [1] or SWEBOK [34]. In our experience, we also found it right for GSwERC [33].

The primary studies aimed at undergraduate curricula were mapped on SE2004 [1]. So, out of total of 70 primary studies, 51 primary studies were in the scope of SE2004 [1], as shown in Fig. 5. The rest of 19 studies were out of the scope of SE2004 [1] because either the studies were not written for Undergraduate Curricula (in case of studies for graduate curricula etc.) such as [11], [17] or the studies were intended to facilitate the instructors e.g. [41], [42]. We also found two studies which were written for an undergraduate SE degree program, on the topics of "Service Engineering and Management' [43] and "Computer Vision for robotic telemedicine cluster system" [44], but no Knowledge Areas (KAs) of SE2004 [1] matched these topics. Again these studies could not be mapped to SE2004 [1].

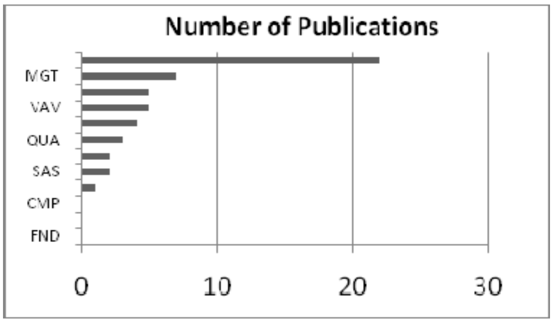

Fig. 5 SE2004 Knowledge Areas covered by our Primary Studies the meanings of acronyms used in Fig. 5 are given below in TABLE IV. From Fig. 5, we found that majority of papers have collectively cover all KAs of SE2004 [1]. This is because majority of our primary studies focused on curriculum revision, such as [45], [46] or applied various learning approaches for teaching SE e.g. [47], [48] thereby, encompassing all KAs of SE2004 [1], rather than focusing on only one KA of SE 2004 [1]

By looking at individual SE2004 [1] KAs in Fig. 5, that much of the studies fall under SW MGT KA, which is comprised of 7 publications. This is because a considerable amount of work has been done to improve student's projects; both capstone and course projects such as [15], [49]. Then a little work has also been done on other KAs of SE2004 [1] as shown in Fig. 5, except for 'Computing Essentials', 'Software Evolution' and 'Mathematical and Engineering Fundamentals'. There is a possibility that a little work might have been done on these KAs, in the papers covering all knowledge areas of SE2004 [1], however, no paper was explicitly based on these KAs. Therefore, we have identified them as the most neglected KA's of SE2004 [1] in SE educational research.

TABLE IV
ACRONYMS OF SE2004 KNOWLEDGE AREAS

Acronyms	Meanings	Acronyms	Meanings
All KAs	Covering all Areas of SE2004 [1]	PRO	Software Process
MGT	Software Management	SAS	System and Application Specialties
PRF	Professional Practice	MAA	Software Modeling and Analysis
VAV	Software Verification and Validation	CMP	Computing Essentials
DES	Software Design	EVO	Software Evolution
QUA	Software Quality	FND	Mathematical and Engineering Fundamentals

Further, we mapped the primary studies written for graduate curricula on GSwERC [33]. Out of 70 primary studies, 38 primary studies were in the scope of GSwERC [33], as shown below in Fig. 6. The rest 32 primary studies were out of the scope of GSwERC [33] because those primary studies were not written for graduate curricula (in case of studies for undergraduate curricula etc.) or the papers were written to facilitate the instructors while teaching SE, e.g. [50], [51].

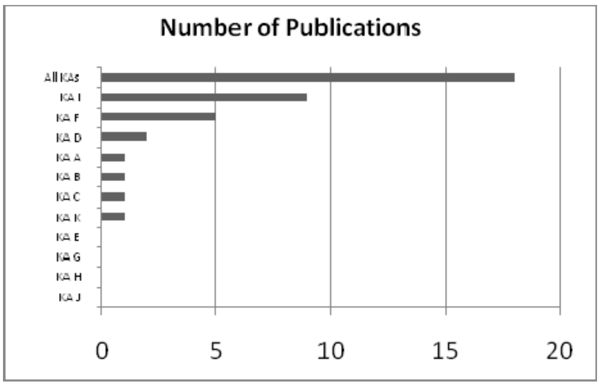

Fig. 6 GSwERC Knowledge Areas covered by our Primary Studies

The meanings of acronyms used in Fig. 6 are given below in Table V. As in the case of SE2004 [1] Fig. 6 also shows that collectively, all KAs of GSwERC [33] have been covered more as compared to individual KA's of GSwERC. It is because of the same reasons mentioned above i.e. due to focus on curriculum revision for example [22], [52] or studies applying various learning approaches such as [18], [53] for teaching SE. By looking at individual GSwERC KAs, we see much work on SW ENG MGT (KA I) same as we saw in SE2004, which is comprised of 9 publications. The work done on KA I is based on various themes such as on Distributed Software Development (DSD) [10], [11], teaching leadership [54], Software Management Master's Program [17].

However, a little work has also been done on other KAs of

GSwERC except for 'Software Construction', 'Software Maintenance', 'Configuration Management' and 'Software Engineering Process'. There is a possibility that a little work might have been done on these KAs, in the papers covering all knowledge areas of GSwERC but specifically, no paper was based on these KAs. Therefore, we have identified them as the most neglected KA's of GSwERC in SE educational research.

TABLE V
ACRONYMS OF GSWERC KNOWLEDGE AREAS

Acronyms	Meanings	Acronyms	Meanings
All KAs	Covering all Knowledge Areas of GSwERC [33]	KA C	Requirements Engineering
KA I	Software Engineering Management	KA K	Software Quality
KA F	Testing	KA E	Software Construction
KA D	Software Design	KA G	Software Maintenance
KA A	Ethics and Professional Conduct	KA H	Configuration Management
KA B	System Engineering	KA J	Software Engineering Process

Lastly, we mapped all the primary studies on SWEBOK [34], irrespective of the degree programs. It is shown in Fig. 7 that out of 70 primary studies, 57 primary studies covered the knowledge areas of SWEBOK. The rest 13 primary studies did not cover the KAs of SWEBOK because those primary studies were out of the scope of SWEBOK in concepts like Professional Practice [12] or in topic like 'Service Engineering and Management' [43], or the papers that were written to facilitate the instructors while teaching SE such as [41], [51].

The acronyms used in Fig. 7 are explained in Table VI. As shown in Fig. 7 a large number of primary studies have collectively covered the all KAs of SWEBOK. This is similar to the case when we mapped the primary studies on SE2004 and GSwERC, as discussed above. We observed a considerable amount of work (f=14) on SW ENG MGT (KA G). All such primary studies were related to develop software engineering management skills in students through various types of projects such as [9], [21]. Furthermore, we see a little bit work on other KAs of SWEBOK [34] except in; 'Software Construction', 'Software Maintenance' and 'Software Configuration Management'. There is a possibility that a little work might have been done on these KAs, in the papers covering all knowledge areas of SWEBOK [34] but specifically, no paper was based on these KAs. Therefore, we have identified them as the most neglected KA's of SWEBOK [34] in SE educational research.

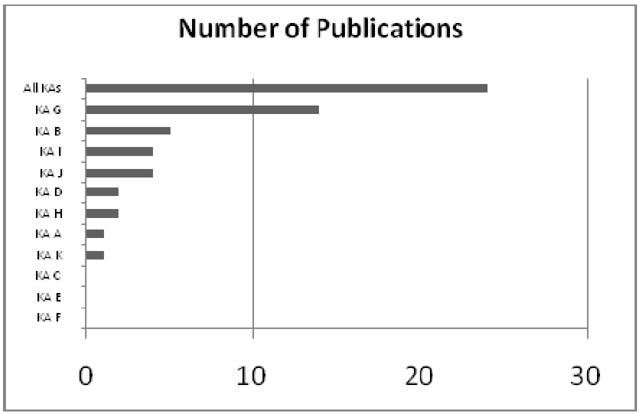

Fig. 7 SWEBOK Knowledge Areas covered by our Primary Studies

TABLE VI
ACRONYMS OF SWEBOK KNOWLEDGE AREAS

Acronyms	Meanings	Acronyms	Meanings
All KAs	Covering all Knowledge Areas of SWEBOK [34]	**KA H**	Software Engineering Process
KA G	Software Engineering Management	**KA A**	Software Requirements
KA B	Software Design	**KA K**	KAs of the Related Disciplines
KA I	Software Engineering Tools and Methods	**KA C**	Software Construction
KA J	Software Quality	**KA E**	Software Maintenance
KA D	Software Testing	**KA F**	Software Configuration Management

After trying to map all our primary studies on SE2004 [1], GSwERC [33] or SWEBOK [34], we concluded that there lies a gap between the Knowledge Areas of reference curriculum; SE2004 [1], GSwERC [33] and SWEBOK [34], and the research done on Software Engineering education. So, the results of our mapping study may be used for the possible revision in the knowledge areas of SE2004 [1], GSwERC [33] and SWEBOK [34].

A. Analysis of Data Gathered in Research Questions

Our first SE educational map, as depicted in Fig. 1, shows the categorization of our primary studies, according to the framework provided by Ellis *et al* [4]. It is shown in Fig. 1 that an extensive amount of research has been done on ITM (f=22). One reason of it might be the result of researchers following the curriculum guideline of ACM/IEEE-CS SE2004 [1], which suggested that SE education should consider other variety of teaching and learning approaches instead of relying only on lecture-based format. As SE is an evolving field, so there is a chance that courses or curriculum become obsolete over the time, therefore instructors and institutions must regularly update their curriculum [1]. It was encouraging to note that a significant number of primary studies aimed to improve C&EM (f=16).

The task force of computing curricula 2001 [55] believes that apart from traditional lecture-oriented format, other styles in teaching such as educational technology should also be considered. A considerable amount of our primary studies focused on how to improve SE education, by using ET (f=14), as shown in Fig. 1 within a typical software development environment, Professional Practice is the understanding and positive reception of the importance of negotiation, a positive communication with stakeholders and is the attitude of effective leadership and work habits [1]. In order to make SE realistic, emphasis should be given on realistic project activities [16]. Despite of this importance of PP (f=9) and PBSE (f=8), comparatively, we found a small number of contributions in these two categories. The category with a minimal research is 'Student Learning an Assessment'. Our second SE educational map, depicted in Fig. 2, shows the names of those publication channels, which contributed more in SE Educational research. It is shown in Fig. 2 that CSEET was the most dominating forum in publishing the SE educational research. After that, ICSE, ICETC and IEEE Trans.Educ published a fair amount of research. ICEIT, SIGCSE, TEE, ICCSE, SPLASH and WCCCE also had their contribution by having thrice or twice publications on SE education.

Our third SE educational map, depicted in Fig. 3, shows that out of 23 participating countries, 11 countries participated more in SE Educational research. The rest 12 countries contributed with a single paper, each. It is depicted in Fig. 3, that USA and China are leading the SE educational research. The more detail on these participating countries is discussed above in section III.

V. Study Limitation and Threats to Validity

The main limitation of our study is the biasness in the selection of relevant literature based on our search strategy. This also includes the time period, we selected for this study. We may have missed some relevant studies but our rigorous search strategy should have accumulated a reasonable sample. Besides these limitations the outcome of the study proved useful for assessing the feasibility of a complete SLR on SEE.

VI. Conclusion

We have performed a mapping study to present a broad view of current advancements taken place to improve Software Engineering education. The objective of conducting this study is also to check the feasibility of conducting SLR on SEE. Based on the suggestions of our experts and on the experience we gained in our pilot study, we developed the search strategy for our mapping study. Out of 1589 studies, we found a total of 70 relevant studies. These 70 primary studies were used as basis to analyze and classify the SE educational research. We classified our primary studies in terms of an SE educational framework, the top publication channels and the top countries which are leading the SE educational research. With respect to SE educational framework, an extensive amount of research has been done on 'Innovative Teaching Methods' and 'Curriculum and Education Management', indicating the focus of researchers to break the impasse in the current SEE practices. A considerable amount of work has been done on 'Educational Technology', 'Professional Practice' and 'Project-based Software Engineering' but a minimal research was found in the category; 'Student Learning and Assessment'.

CSEET was the top publication channel in SE educational research. Out of total of 28 conferences and 1 Journal, we found a fair amount of research in ICSE, ICETC and IEEE Trans.Educ. We also noticed that USA and China are leading the SE educational research, out of 23 participating countries, in total.

We further mapped our primary studies on three reference curriculums; SE2004, GSwERC and SWEBOK. We found that current knowledge areas of these reference curriculums are not covering the diverse aspects of SE educational research. Hence, the results of our mapping study may be used for the possible revision in the knowledge areas of SE2004, GSwERC and SWEBOK. We also mapped our primary studies on the '4-tier taxonomy of software exporting nations, where we found that mostly the top software exporting nations were leading the SE educational research. We highlight this point in order to motivate the educationists, researchers and practitioners of Emerging, Infant stage and Non-Competing Software Exporting nations to improve their SE education also.

The facts and figures provided in our study will help:

Academicians: To gain an understanding of proposed and/or proven practices in software engineering education that could be employed at their own institutions. This will help them in refining and defining their curriculums.

Practitioners: To get benefit from an understanding of the synergies between educational practices and real-world software development. This, in turn, will help Practitioners to guide/suggest academia with further improvements in curriculum in order to produce better software engineers.

Researchers: To get the new research topics for their future research.

Our mapping study can be reproduced with other Point of Views, Research Questions and other limitations, stated above. This mapping study has served as a pilot study for our SLR on SE education (in process), in order to deeply analyze the latest innovations taken place to improve SE education. We end our discussion on mapping study with a hope that those educational institutes who have not started revising their curriculum for betterment may start it from now, in order to produce better software engineers for tomorrow.

Appendix A. Studies Included in the Review

[PS1] F. Alonso, D. Manrique, L. Martínez, and J. M. Viñes, "How Blended Learning Reduces Underachievement in Higher

Education: An Experience in Teaching Computer Sciences," *Education, IEEE Transactions* pp. 1-8, 2010.

[PS 2] T. Astigarraga, E. M. Dow, C. Lara, R. Prewitt, and M. R. Ward, "The Emerging Role of Software Testing in Curricula," in *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, Dublin,Ireland, 2010, pp. 1-13.

[PS 3] R. Bareiss and G. Mercier, "A Graduate Education in Software Management and the Software Business for Mid-Career Professionals," in CSEE&T, Pittsburgh, PA, 2010: pp. Pages.

[PS 4] R. Bareiss and M. Radley, "Coaching Via Cognitive Apprenticeship," in *SIGCSE*, Milwaukee, Wisconsin, USA., 2010, pp. 162-166.

[PS 5] J. Bolinger, K. Yackovich, R. Ramnath, J. Ramanathan, and N. Soundarajan, "From Student to Teacher.Transforming Industry Sponsored Student Projects into Relevant, Engaging, and Practical Curricular Materials," in *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, Dublin,Ireland, 2010, pp. 1-8.

[PS 6] I. Bosnic, I. Cavrak, M. Žagar, R. Land, and I. Crnkovic, "Customers' Role in Teaching Distributed Software Development," in CSEE&T, Pittsburgh, PA, 2010: pp. Pages.

[PS 7] D. Broman, "Should Software Engineering Projects be the Backbone or the Tail of Computing Curricula?," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 153-156.

[PS 8] R. Cardell-Oliver, *et al.*, "Automated Feedback for Quality Assurance in Software Engineering Education," in *ASWEC*, Auckland,Newzeland, 2010, pp. 157-164.

[PS 9] Y. CHEN and L. YE, "Innovation and Practice on CDIO-based Three-dimensional Integrated Curriculum Model," in *ICEIT*, Chongqing ,China., 2010, pp. V2-542 - V542-545.

[PS 10] A. Chidanandan, L. Russell-Dag, C. Laxer, and R. Ayfer, "In Their Words: Student Feedback on an International Project Collaboration," in *SIGCSE*, Milwaukee, Wisconsin, USA, 2010

[PS 11] D. Chudá and D. Rodina, "Automata Simulator," in *CompSysTech*, Sofia, Bulgaria, 2010, pp. 394-399.

[PS 12] P. J. Clarke, A. A. Allen, T. M. King, E. L. Jones, and P. Natesan, "Using a Web-Based Repository to Integrate Testing Tools into Programming Courses," in *SPLASH*, Reno/Tahoe, Nevada, USA, 2010, pp. 193-200.

[PS 13] A. J. Cowling, "Stages in Teaching Formal Methods," in CSEE&T, Pittsburgh, PA, 2010: pp. Pages.

[PS 14] V. Devedˇzic´ and S. s. R. Milenkovic, "Teaching Agile Software Development: A Case Study," *Education, IEEE Transactions*, vol. pp, pp. 1-6, 2010.

[PS 15] M. Devlin and C. Phillips, "Assessing Competency in Undergraduate Software Engineering Teams," in *EDUCON*, Madrid,Spain, 2010, pp. 271-277.

[PS 16] R. G. Epstein, "Workplace Issues in an Undergraduate Software Engineering Course," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 1-8.

[PS 17] J. Fendler and H. Winschiers-Theophilus, "Towards Contextualised Software Engineering Education: An African Perspective," in *ICSE*, Cape Town, South Africa, 2010, pp. 599-606.

[PS 18] G. Finnie and P. Krishnan, "Adding Service Engineering and Management to a Software Engineering Program," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 165-168.

[PS 19] S. R. Foster and R. Ruiz, "Constructivist Pedagogy meets Agile Development – A Case Study," in *ITNG*, Las Vegas, NV, 2010, pp. 1092-1096.

[PS 20] Y. Gao and X. Li, "To Innovate the Teaching of Software Engineering in Order to Train Student's Creative Thinking Idea," in *ICETC*, Shanghai,China, 2010, pp. V2-347 - V342-349.

[PS 21] V. Garousi, "Applying Peer Reviews in Software Engineering Education: An Experiment and Lessons Learned," *Education, IEEE Transactions* vol. VOL. 53, NO. 2,, pp. 182-193, 2010.

[PS 22] D. J. Griffin and P. K. Ryan, "Broadening the Education of Software Engineers – some lessons and pointers," in *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, Dublin,Ireland., 2010, pp. 1-6.

[PS 23] M. Guo, T. Chai, and K. Qian, "Design of Online Runtime and Testing Environment for Instant Java Programming Assessment," in *ITNG*, Las Vegas, NV ,USA, 2010, pp. 1102-1106.

[PS 24] S. Hanenberg, "Faith, Hope, and Love An essay on software science's neglect of human factors," in *OOPSLA*, Reno-Tahoe, Nevada, USA, 2010, pp. 933-946.

[PS 25] O. Hazzan and Y. Dubinsky, "A HOT – Human, Organizational and Technological – Framework for a Software Engineering Course," in *ICSE*, Cape Town, South Africa, 2010, pp. 559-566.

[PS 26] O. Hazzan and Y. Dubinsky, "Students' Cooperation in Teamwork: Binding the Individual and the Team Interests," in *SPLASH*, Reno/Tahoe, Nevada, USA, 2010

[PS 27] R. Hoar, "Yes they can! Second-year Student Software Development: A Case Study," in *WCCCE*, Kelowna, Canada, 2010, pp. 1-6.

[PS 28] A. Humayun, W. Basit, G. A. Farrukh, F. Lodhi, and R. Aden, "An Empirical Analysis of Team Review Approaches for Teaching Quality Software Development," in *ICSE*, Cape Town, South Africa, 2010, pp. 567-575.

[PS 29] J. M. Hunt and J. D. McGregor, "Component Based Software Engineering across the Curriculum," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 9-16.

[PS 30] Y. Jia, "Improving Software Engineering Courses with Case Study Approach," in *ICCSE*, Hefei ,China, 2010, pp. 1633-1636.

[PS 31] S. Kang, I.-Y. Ko, J. Baik, H. Choi, and D. Lee, "KAIST-CMU MSE Program – the Past and the Future," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 49-56.

[PS 32] R. Kaplan and J. Tan, "Real World Experiences in a Software Engineering Course," in *SIGITE*, Midland, Michigan, USA., 2010, pp. 89-93.

[PS 33] E. P. Katz, "Software Engineering Practicum Course Experience," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 169-172.

[PS 34] J. Kietzmann and H. H. Tsang, "Minding the Gap: Bridging Computing Science and Business Studies with an Interdisciplinary Innovation Challenge," in *WCCCE*, Kelowna, British Columbia, Canada, 2010, pp. 1-5.

[PS 35] V. Kulkarni, C. Scharff, and O. Gotel, "From Student to Software Engineer in the Indian IT Industry: A Survey of Training," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 57-64.

[PS 36] C. Y. Lester, "Training undergraduate students in user-centered design," in *ACHI*, Saint Maarten, 2010, pp. 185-190.

[PS 37] M. W. Lew, T. B. Horton, and M. S. Sherriff, "Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 121-128.

[PS 38] Z. Liang, BinYu, and ShuiWang, "Curriculum Programme of Career-oriented Java Specialty Guided by Principles of Software Engineering," in *ICETC*, Shanghai,China, 2010, pp. V1-592 - V591-595.

[PS 39] H. Liu, F.-C. Kuo, and T. Y. Chen, "Teaching an End-User Testing Methodology," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 81-88.

[PS 40] S. Liu, J. Tang, C. Deng, X.-F. Li, and J.-L. Gaudiot, "RHE: A JVM Courseware," *Education, IEEE Transactions*, vol. pp, pp. 1-8, 2010.

[PS 41] R. Llamosa-Villalba and S. E. M. Aceros, "Process Management Model for Higher Education. Improvement of Educational Programs in Software Quality," in EDUCON, Madrid,Spain, 2010: pp. Pages.

[PS 42] M. M. Luis, L. Guillermo, R.-D. A, and C. J. R., "Experiences in Software Engineering Courses Using Psychometrics with RAMSET," in *ITiCSE*, Bilkent, Ankara, Turkey, 2010, pp. 244-248.

[PS 43] J. Magenheim, *et al.*, "Competencies for Informatics Systems and Modeling. Results of Qualitative Content Analysis of Expert Interviews," in *EDUCON*, Madrid 2010, pp. 513-521.

[PS 44] S.-H. Mirian-Hosseinabadi, Z. Aghakasiri, A. Sadeghi, P. Delfani, and M. Ghandehari, "Emphasizing Experiences in Teaching Software Engineering Courses," in *ICETC*, Shanghai ,China, 2010, pp. V2-149 - V142-153.

[PS 45] M. Nordio, R. Mitin, and B. Meyer, "Advanced Hands-on Training for Distributed and Outsourced Software Engineering," in ICSE, Cape Town, South Africa, 2010: pp. Pages.

[PS 46] W. Pádua, "Measuring complexity, effectiveness and efficiency in software course projects," in *ICSE*, Cape Town, South Africa, 2010, pp. 545-554.

[PS 47] F. Paulisch and P. Zimmerer, "A role-based qualification and certification program for software architects: An experience report from Siemens," in *ICSE*, Cape Town, South Africa, 2010, pp. 21-27.

[PS 48] R. Razali and M. Chitsaz, "Cases Development for Teaching Software Engineering," in *ICETC*, Shanghai,China, 2010, pp. V2-121 - V122-125.

[PS 49] M. Ricken and R. Cartwright, "Test-First Java Concurrency for the Classroom," in *SIGCSE*, Milwaukee, WI, USA, 2010, pp. 219-223.

[PS 50] C. S. C. Rodrigues, "VisAr3D: An Approach to Software Architecture Teaching Based on Virtual and Augmented Reality," in *ICSE*, Cape Town, South Africa, 2010, pp. 351-352.

[PS 51] M. S. Roslina and D. A. Noraziah, "Venn Requirement Language for User Requirement," in *ICECT*, Kuala Lumpur,Malaysia, 2010, pp. 223-227.

[PS 52] L. A. Sudol and C. Jaspan, "Analyzing the Strength of Undergraduate Misconceptions About Software Engineering," in *ICER*, Aarhus, Denmark, 2010, pp. 31-39.

[PS 53] G. Taran and J. Hartje, "Software Engineering Leadership: A Student Initiative to Promote Change Agents in Industry," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 173-176.

[PS 54] J. B. Thompson, "Why Better Industrial/Academic Links Are Needed If There Is To Be An Effective Software Engineering Workforce," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 105-112.

[PS 55] P. Tiejun, Z. Leina, Z. Yanhong, and F. Chengbin, "Co-operative Studies Supporting System for Software Engineering," in *SEDM*, Chengdu,China, 2010, pp. 1-4.

[PS 56] P. Tiejun, F. Yuefeng, Z. Leina, Y. Xiaoyan, and L. Yunpeng, "Research of High-quality Innovative and Pioneering Undergraduate Training Model of Software Engineering," in *ICETC*, Shangha,China, 2010, pp. V3-203 - V203-207.

[PS 57] N. Tillmann, J. d. Halleux, and T. Xie, "Parameterized Unit Testing: Theory and Practice," in *ICSE*, Cape Town, South Africa, 2010, pp. 483-484.

[PS 58] M. Usman, J. Khan, M. Hardas, and N. Ikram, "Pedagogical and Structural Analysis of SE Courses," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 136-143.

[PS 59] H. Wang, T. Wang, and S. Tan, "The Reform and Exploration on Education Model of Software Engineering Talents," in *ICCSE*, Hefei, China, 2010, pp. 1932-1934.

[PS 60] Y. Wang, HongyuDu, and Y. Hao, "A Case Study of Problem-Based Learning Instruction Design under Web-Based Environment," in *ICETC*, Shanghai, China, 2010, pp. VI-586-VI-559I.

[PS 61] Z. Wang and C. Yang, "Research on Case Learning System for Engineering Subject ----Software Engineering as an Example," in *CIT*, Bradford, United Kingdom 2010, pp. 2101-2104.

[PS 62] K. Z. Watkins and T. Barnes, "Competitive and Agile Software Engineering Education," in *SoutheastCon*, Concord, NC, 2010, pp. 111-114.

[PS 63] XiaoyongLU, HuizhenXUE, Haoouyang, and M. TU, "Practice of Using Suite of Teaching to Build the Students' Forming Capacity of Software Engineering," in *ICCSNA*, Hong Kong, 2010, pp. 308-311.

[PS 64] J. Xueyun and C. Zihui, "A New Way to Software Engineering Education," in *ICEIT*, Chongqing,China, 2010, pp. V2-1 -- V2-3.

[PS 65] S. S. Yadav and J. Xiahou, "Integrated Project Based Learning in Software Engineering Education," in *ICENT*, Qinhuangdao, China, 2010, pp. 34-36.

[PS 66] T. Yamakami, "Cross-boundary Software Engineering: Implications of Engineering Paradigm Shift," in *ICIS*, Chengdu, China 2010, pp. 536-540.

[PS 67] L. Ying, Z. Yan, and L. Tong, "Strategies for promoting better interaction between higher education and IT industry," in *ICEE*, Guangzhou,China, 2010, pp. 5520-5522.

[PS 68] J. Zhang and J. Li, "Teaching Software Engineering Using Case Study," in *ICBECS*, Wuhan,China., 2010, pp. 1-4.

[PS 69] H. Ziv and S. Patil, "Capstone Project: From Software Engineering to “Informatics”," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 185-188.

[PS 70] W. Zongjiang, "The Study of Network Teaching System Development Based Software Engineering," in *ICEIT*, Chongqing,China, 2010, pp. V2-100 - V102-103.

ACKNOWLEDGMENT

The authors would like to thank Mahmood Niazi, Oscar Dieste and one another anonymous reviewer for their valuable comments, at the earlier stages of our work. The authors are also grateful to Fabio Q.B. da Silva, Erran Carmel and another anonymous reviewer for guiding us in the later stages of our mapping study.

REFERENCES

[1] "IEEE & ACM JTFCC,Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," IEEE & ACM;The Joint Task Force on Computing Curricula 23 August 2004.

[2] P. J. Clarke, A. A. Allen, T. M. King, E. L. Jones, and P. Natesan, "Using a Web-Based Repository to Integrate Testing Tools into Programming Courses," in *SPLASH*, Reno/Tahoe, Nevada, USA, 2010, pp. 193-200.

[3] M. Ardis, *et al.*, "Advancing Software Engineering Professional Education," in *accepted for publication in IEEE Software*, 2010, pp. 1-11.

[4] H. J. C. Ellis, S. A. Demurjian, and J. F. Naveda, *Software Engineering: Effective Teaching and Learning Approaches and Practices*. New York: Information Science Reference, 2008, pp. 1-404.

[5] S. Honiden, Y. Tahara, N. Yoshioka, K. Taguchi, and HironoriWashizaki, "Top SE: Educating Superarchitects Who Can Apply Software Engineering Tools to Practical Development in Japan," in *Proceedings of the 29th international conference on Software Engineering*, 2010, pp. 708-718.

[6] V. Kulkarni, C. Scharff, and O. Gotel, "From Student to Software Engineer in the Indian IT Industry: A Survey of Training," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 57-64.

[7] M. Shaw, "Software Engineering Education: A Roadmap," in Future of Sofware Engineering, Limerick, Ireland, 2000: pp. Pages.

[8] Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Software Engineering Group,School of Computer Science and Mathematics,Keele University & Department of Computer Science University of Durham ,UK, EBSE Technical Report EBSE-2007-01, 9 July 2007.

[9] D. Broman, "Should Software Engineering Projects be the Backbone or the Tail of Computing Curricula?," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 153-156.

[10] M. Nordio, R. Mitin, and B. Meyer, "Advanced Hands-on Training for Distributed and Outsourced Software Engineering," in ICSE, Cape Town, South Africa, 2010: pp. Pages.

[11] I. Bosnic, I. Cavrak, M. Žagar, R. Land, and I. Crnkovic, "Customers' Role in Teaching Distributed Software Development," in CSEE&T, Pittsburgh, PA, 2010: pp. Pages.

[12] R. G. Epstein, "Workplace Issues in an Undergraduate Software Engineering Course," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 1-8.

[13] V. Garousi, "An Open Modern Software Testing Laboratory Courseware " in *23rd IEEE Conference on Software Engineering Education and Training*, Pittsburgh, Pennsylvania 2010, pp. 177-184.

[14] A. J. Cowling, "Stages in Teaching Formal Methods," in CSEE&T, Pittsburgh, PA, 2010: pp. Pages.

[15] S. S. Yadav and J. Xiahou, "Integrated Project Based Learning in Software Engineering Education," in *ICENT*, Qinhuangdao, China, 2010, pp. 34-36.

[16] S. Hadjerrouit, "Learner-Centered Web-Based Instruction in Software Engineering," *IEEE TRANSACTIONS ON EDUCATION*, vol. 48, NO. 1, pp. 99-104, 2005.

[17] R. Bareiss and G. Mercier, "A Graduate Education in Software Management and the Software Business for Mid-Career Professionals," in CSEE&T, Pittsburgh, PA, 2010: pp. Pages.

[18] R. Bareiss and M. Radley, "Coaching Via Cognitive Apprenticeship," in *SIGCSE*, Milwaukee, Wisconsin, USA., 2010, pp. 162-166.

[19] M. W. Lew, T. B. Horton, and M. S. Sherriff, "Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 121-128.

[20] H. Liu, F.-C. Kuo, and T. Y. Chen, "Teaching an End-User Testing Methodology," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 81-88.
[21] E. P. Katz, "Software Engineering Practicum Course Experience," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 169-172.
[22] J. Fendler and H. Winschiers-Theophilus, "Towards Contextualised Software Engineering Education: An African Perspective," in *ICSE*, Cape Town, South Africa, 2010, pp. 599-606.
[23] J. Bailey, *et al.*, "Evidence relating to Object-Oriented software design: A survey," in First International Symposium on Empirical Software Engineering and Measurement, 2007: pp. Pages.
[24] "Evidence-Based Software Engineering", Accessed on 5 March 2012, http://www.dur.ac.uk/ebse/templates.php
[25] E. Y. Nakagawa., D. Feitosa., and K. R. Felizardo, "Using Systematic Mapping to Explore Software Architecture Knowledge," in SHARK, cape town, south Africa, 2010: pp. Pages.
[26] "EndNote Software", Accessed on 19 March 2012, http://www.endnote.com/enhome.asp
[27] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in 12th International Conference on Evaluation and Assessment in Software Engineering Italy, 2008: pp. Pages.
[28] H. J. C. Ellis, R. A. Morelli, T. R. d. Lanerolle, and G. W. Hislop, "Holistic Software Engineering Education Based on a Humanitarian Open Source Project," in *CSEE&T*, 2007, pp. 1-9.
[29] H. J. C. Ellis, A. Moreno, N. R. Mead, and S. B. Seidman, "Reeducation to Expand the Software Engineering Workforce: Successful Industry/University Collaborations," Carnegir Mellon Software Engineering Institute, Pittsburgh CMU/SEI-2002-SR-001, July 2002 2002.
[30] H. J. C. Ellis and G. W. Hislop, "Techniques for Providing Software Engineering Education to Working Professionals," in 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA, 2004: pp. Pages.
[31] A. J. Budd and H. J. C. Ellis, "Spanning the Gap Between Software Engineering Instructor and Student," in *ASEE/IEEE Frontiers in Education Conference*, 2008, pp. S3H-10-S13H-15.
[32] E. Carmel, "Taxonomy of New Software Exporting Nations " *The Electronic Journal of Information Systems in Developing Countries*, pp. 1-6, 2003.
[33] "Graduate Software Engineering Reference Curriculum [GSwERC] v 0.5," Stevens Institute of Technology Version 0.50, October 31, 2008 2008.
[34] A. Abran, P. Bourque, and L. L. Tripp, "Guide to the Software Engineering Body of Knowledge (SWEBOK)," IEEE Computer Society Professional Practices Committee 2004.
[35] L. Ying, Z. Yan, and L. Tong, "Strategies for promoting better interaction between higher education and IT industry," in *ICEE*, Guangzhou, China, 2010, pp. 5520-5522.
[36] L. Kuivanen and N. Nahar, "Vietnam as an emerging destination for offshore outsourcing of software development for finnish companies: A conceptual perspective," in *Portland International Conference on Management of Engineering & Technology, (PICMET).* , Portland,OR, 2009, pp. 1110 - 1123.
[37] E. Carmel and P. Tjia, *Offshoring information technology: sourcing and outsourcing to a global workforce*: Cambridge University Press, 2005.
[38] B. Kitchenham, *et al.*, "Systematic literature reviews in software engineering – A tertiary study," *Information and Software Technology*, pp. 792-805, 2010.
[39] F. Q. B. d. Silva, *et al.*, "Six years of systematic literature reviews in software engineering: An updated tertiary study," *Information and Software Technology*, pp. 1-15, 2011.
[40] R. Llamosa-Villalba and S. E. M. Aceros, "Process Management Model for Higher Education. Improvement of Educational Programs in Software Quality," in EDUCON, Madrid, Spain, 2010: pp. Pages.
[41] M. Devlin and C. Phillips, "Assessing Competency in Undergraduate Software Engineering Teams," in *EDUCON*, Madrid, Spain, 2010, pp. 271-277.
[42] M. Usman, J. Khan, M. Hardas, and N. Ikram, "Pedagogical and Structural Analysis of SE Courses," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 136-143.
[43] G. Finnie and P. Krishnan, "Adding Service Engineering and Management to a Software Engineering Program," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 165-168.
[44] Z. Chaczko and R. Braun, "Teaching Computer Vision for Telemedicine Systems using OpenCV," in *ITHET*, Cappadocia,Turkey, 2010, pp. 17-20.
[45] J. Bolinger, K. Yackovich, R. Ramnath, J. Ramanathan, and N. Soundarajan, "From Student to Teacher. Transforming Industry Sponsored Student Projects into Relevant, Engaging, and Practical Curricular Materials," in *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, Dublin,Ireland, 2010, pp. 1-8.
[46] P. Tiejun, F. Yuefeng, Z. Leina, Y. Xiaoyan, and L. Yunpeng, "Research of High-quality Innovative and Pioneering Undergraduate Training Model of Software Engineering," in *ICETC*, Shangha,China, 2010, pp. V3-203 - V203-207.
[47] F. Alonso, D. Manrique, L. Martínez, and J. M. Viñes, "How Blended Learning Reduces Underachievement in Higher Education: An Experience in Teaching Computer Sciences," *Education, IEEE Transactions* pp. 1-8, 2010.
[48] S.-H. Mirian-Hosseinabadi, Z. Aghakasiri, A. Sadeghi, P. Delfani, and M. Ghandehari, "Emphasizing Experiences in Teaching Software Engineering Courses," in *ICETC*, Shanghai ,China, 2010, pp. V2-149 - V142-153.
[49] A. Chidanandan, L. Russell-Dag, C. Laxer, and R. Ayfer, "In Their Words: Student Feedback on an International Project Collaboration," in *SIGCSE*, Milwaukee, Wisconsin, USA, 2010
[50] W. Pádua, "Measuring complexity, effectiveness and efficiency in software course projects," in *ICSE*, Cape Town, South Africa, 2010, pp. 545-554.
[51] M. M. Luis, L. Guillermo, R.-D. A, and C. J. R., "Experiences in Software Engineering Courses Using Psychometrics with RAMSET," in *ITiCSE*, Bilkent, Ankara, Turkey, 2010, pp. 244-248.
[52] S. Kang, I.-Y. Ko, J. Baik, H. Choi, and D. Lee, "KAIST-CMU MSE Program – the Past and the Future," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 49-56.
[53] Y. Jia, "Improving Software Engineering Courses with Case Study Approach," in *ICCSE*, Hefei ,China, 2010, pp. 1633-1636.
[54] G. Taran and J. Hartje, "Software Engineering Leadership: A Student Initiative to Promote Change Agents in Industry," in *CSEE&T*, Pittsburgh, PA, 2010, pp. 173-176.
[55] "Computing Curricula 2001 Computer Science," The Joint Task Force on Computing Curricula, IEEE Computer Society & Association for Computing Machinery 2001.