Challenges and Recommendations for the Design and Conduct of Global Software Engineering Courses: A Systematic Review Protocol

Sarah Beecham, Tony Clear, John Barr & John Noll

Technical Report No. Lero_TR_2015_01

ITiCSE Working Group One: Review published: Clear, T., S. Beecham, J. Barr, M. Daniels, R. McDermott, M. Oudshoorn, A. Savickaite and J. Noll (2015). Challenges and Recommendations for the Design and Conduct of Global Software Engineering Courses: A Systematic Review. ACM Proceedings of the Working Group Reports of the 2015 on Innovation & Technology in Computer Science Education Conference (ITiCSE'15), Vilnius, Lithuania, ACM. DOI: http://dx.doi.org/10.1145/2858796.2858797

Contact Information:
Sarah Beecham
Lero – University of Limerick
sarah.beecham@lero.ie

Protocol for a SLR of Teaching Global Software Engineering (GSE)

Preamble

This SLR we are conducting traverses the many options available to Computer Science (CS) educators teaching CS courses involving global collaboration. The challenges and solutions in conducting global software engineering courses will be addressed. While there is a rich source of literature covering this topic, there is limited consolidated guidance available for CS educators wishing to implement a global course, in collaboration with other institutions. So building upon the existing knowledge in the literature in the area will help to produce a report that will serve as a broad ranging resource for global software engineering educators.

The SLR focusses on two areas:

1. Learning GSE Theory: Developing courses based on GSE theory. I.e. How to teach students about developing software across multi-site teams (to include things like cultural training – i.e. how to build trust amongst a team that hasn't met face to face, etc.).

AND

2. Learning GSE by doing: Developing courses that show how to apply GSE methods in the classroom. E.g. where students develop software in multi-site teams (where the software developed is not really the focus, but 'how' to develop the software is what we would be looking at).

We also include studies that take a hybrid approach by including a combination of theory and practice. I.e. research that presents experiences of running hybrid courses aimed at developing student capabilities in working as global professionals which have varying degrees of cross-site collaboration, and theory/practice balance.

1. Background

The proposed systematic literature review is concerned with a crucial area of software engineer education and training: – *how to teach global software engineering methods to students before they enter the workplace?* While there is increasing recognition that GSE requires special treatment, and that students entering the workplace are likely to find themselves working in distributed teams, apart from the start of the art review provided by [5], no review found in the GSE education literature has been undertaken to bring together the combined knowledge into a set of educator specific recommendations on the topic..

GSE is increasingly cited as becoming the norm [1, 2, 3, 4]. Students studying SE are very likely to find themselves working in multi-site teams. Yet GSE projects often fail to realise hoped-for advantages such as higher productivity through hiring highly skilled engineers from countries with competitive labour rates. The challenge of developing software across global distance (temporal, geographic and cultural), is complex. Many organisations are realising that they need to invest in cultural training to improve team collaboration [5, 6]. If educators of the future workforce can pre-empt this need, the new tranche of engineers will be better equipped for the unique challenges imposed on them by working in multi-site teams.

The studies in this area suggest that conventional approaches to teaching SE are outdated.

The literature is presenting mixed messages. The balance between developing students' with strong technical skills and augmenting those with a broader set of professional capabilities has long been a source of tension in the academy. Traditionally these challenges in computer science and software engineering programmes have been addressed through capstone courses and internship models [7, 8]. However with the rise of globalisation and the concomitant changes in the working environment for professional software engineers [9], new approaches are needed, and a number of collaborative software engineering programmes have arisen in response [9, 10,11, 12, 13, 14, 15]. These initiatives have mostly been pioneering and relatively discrete, and have represented non-trivial commitments for the participating institutions. Some of the collaborations however have been long lived e.g. [11, 16, 22]. In courses of this nature a number of issues inevitably arise from the challenges of the distances posed by time, space, organisational, linguistic and cultural boundaries [16, 17, 18, 19, 20, 21].

Managing ambiguity and complexity are key capabilities that students must develop if they are to have an education that endures [16, 18, 21]. Since we do not have all the answers for doing this well, it therefore behoves us to continue to develop models, practices and strategies that will serve both students and educators, as well as the profession. A starting point for capturing these methods is to identify what has worked well in GSD teaching as reported in the literature. Also, of interest to educators is an understanding of known obstacles to teaching GSD to students in a university setting.

2. Research Questions

We considered whether our general research question, "What are the key approaches to designing and conducting GSE courses?" is suitable for investigation by systematic review. *Prima facie* this question does not closely match the type suggested by Kitchenham and Charters (2007) where the emphasis is on assessing how technology is adopted in/affects software engineering. Our work perhaps relates more closely to the root of the guidelines provided by the medical literature. We can adapt a medical theme,
"Assessing the economic value of an intervention or procedure", to "Assessing the [economic] value of applying recommended design approaches to global software engineering courses". In our case we can interpret "economic" in terms of a student's readiness to work in GSE.

Initial research shows very little work in the area of the economics of education in global software engineering. Therefore, to answer our key research question in terms of the value GSE courses bring to the student and the workplace we pose two sub-questions:

RQ1: What are the challenges in delivering GSE courses to SE Students?
RQ2: What are the recommendations for delivering GSE courses to SE Students?

We need to address both these questions as there may be barriers (RQ1) to implementing certain recommended practices (RQ2). Solutions (RQ2) need to be in context with any known constraints (RQ1). The context of the education setting is Higher/Third tier. The recipients of these courses can therefore be full time students (with no industrial experience), or Software engineers (professionals), participating in Higher Education or related training.

2.2 Constructing Search terms

The following details of the population, intervention, outcomes, and experimental designs of interest to the review will form the basis for the construction of suitable search terms later in the protocol (Section 3.1).

Population: Software Engineer Students (based in tertiary ed./university settings)
Intervention: GSE teaching and learning approaches
Outcomes of relevance: Evidence of learning, Cost Saving, Relevance to workplace, sustainability/institutionalisation of the initiative.
Experimental design: Empirical studies, theoretical studies, expert observation, experience reports – showing 'how' courses are delivered (e.g. classroom based, or problem based learning, assessment schemes etc.).

Breaking down research question 2 to include these details:
RQ2: What are the recommendations for delivering successful GSE courses to SE Students?
[What are the recommendations] ***INTERVENTION***
[for delivering] ***EXPERIMENTAL DESIGN***
[successful GSE courses] to ***OUTCOMES OF RELEVANCE*** **[Software Engineer Students]** ***POPULATION***

Although the **experimental design** is included in the research question we are 'open' to the types of study we include as we don't want to preclude any new method. This area is multi-disciplinary since GSE courses require both a theoretical (framework), and practical empirical evidence of how theory is applied in practice.

Empirical studies include ethnographic observational studies, action research, questionnaires, individual interviews and focus groups. *Theoretical studies* are those not based on an experiment or direct observation, for example when an expert makes observations and draws on some of the educational literature and theoretical frameworks from related foundation disciplines such as psychology and sociology and organisational behaviour. Until the literature review is complete, it is not possible to predict whether there is a general approach to recognising barriers and solutions to GSE teaching approaches. Appendix A (inclusion criteria) relates to experimental design. All papers in our review will categorise the experimental design as reported in our spreadsheet metadata under 'Type of Study', see section 4.1.2.

On completion of the systematic literature review, this experimental design categorisation will allow us to identify whether there is a standard study approach, and will also allow us to conduct sensitivity analyses based on experimental design.

2.3 Study Type (according to Valentines' taxonomy (Valentine 2004))

Since we are looking mainly at research undertaken in a classroom / education setting, we also use Valentine's definitions of study types. Valentine observed that existing classifications of study types did not cater for the range of studies undertaken in educational research. A six-fold taxonomy to classify the type of articles found in Educational Research. Valentine suggests that we do not need a strictly quantified, statistical model to prove significant educational results. As a result he set "as inclusive (and yet reasonable) a bar as possible for this category" and settled on a simple rubric. See Appendix D for definitions.

3. Search Strategy

3.1. Identifying search terms for automated searches

The strategy used to construct search terms is as follows:

a. derive major terms from the questions by identifying the population, intervention and outcome;
b. identify alternative spellings and synonyms for major terms;
c. check the keywords in any relevant papers we already have;
d. when database allows, use the Boolean OR to incorporate alternative spellings and synonyms;
e. when database allows, use the Boolean AND to link the major terms from population, intervention and outcome.

Results for a) – major terms

For clarity, terms for each research question are given separately.

RQ1: Software engineer student, challenges, GSE courses, delivery
RQ2: Software engineer student, recommendations, successful GSE courses

Results for b) – synonyms and alternative spellings for (a)
* = truncation

Software engineer student: (software OR "information technology" OR "information system*" OR comput* OR programming OR programing OR IT OR IS) AND (student OR trainee OR learner)

Challenges: challenge* OR barrier* OR bottleneck OR problem OR issue OR "lessons learned"

Successful: success* OR relevance OR recommend* OR model OR framework OR practice OR strategy

GSE courses: ("Distributed software" OR Multi-site" OR "multi-site" OR "Global Software" OR collaborative OR virtual) AND ("distributed team*" AND (education OR training OR tutorial OR teach*)

Results for c)
We used a wide set of search terms, and captured all known works.

Results for d) and e)
Search Terms will be changed to suit each database. Appendix B provides a lookup table that maps the ACM database to its search strings. As some databases have different syntax and search rules, the example below will often be modified and sometimes simplified (see section 3.2 for list of Databases).

RQ1
((software OR "information technology" OR "information system*" OR comput* OR programming OR programing OR IT OR IS) AND (student OR trainee OR learner)) AND
(challenge* OR barrier* OR bottleneck OR problem OR issue OR "lessons learned")
AND ("distributed software" OR "multi-site" OR "multi-site" OR "global software" OR "distributed team*)
AND (educat* OR train* OR tutorial OR teach* OR course))

RQ2}
((software OR {information technology} OR {information system*} OR comput* OR programming OR programing OR IT OR IS) AND (student OR trainee OR learner)
AND (success* OR relevance OR recommend* OR model OR framework OR practice)
AND ({distributed software} OR {multi-site} OR {multi-site} OR {global software"}) AND {"distributed team*} AND {educat* OR train* OR tutorial OR teach* OR course}))

Using command search in IEEExplore, and searching in metadata - using all keywords listed in this section (a, b, c and d above) produced too many papers and false positives (over 40,000). We therefore paired down the number of options (separated by Boolean OR), to the core words. Our new search string reads:

((((software OR "information technology" OR "information system*" OR comput* OR programming) AND (student OR trainee OR learner)) AND ("distributed software" OR "global software") AND (educat* OR train* OR course))) and refined by
Content Type: Conference Publications Journals & Magazines
Year: 2000-2015

This yielded 545 papers.

The 545 papers were circulated to three key researchers for **validation** and selection based on title and abstract.

3.2 Resources to be searched:

Databases

- IEEE Digital Library (www.computer.org)
- ACM Digital Library (http://portal.acm.org/dl.cfm)
- Scopus (http://www.elsevier.com/solutions/scopus)

Other sources:

International Conference on Global Software Engineering (ICGSE) - key conference for GSE/GSD
International Conference on Innovation & Technology in Computer Science Education (ITiCSE) - key conference for CS & SE Education
Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD) - Workshop for GSD & Teaching

NB: ICGSE proceedings papers are found in IEEEXplore, and ITiCSE papers are accessed via ACM. However, although we have used both IEEEXplore and ACM bibliographic databases in our searches – we limited the papers to those that included our search terms. To ensure we don't miss any papers that don't conform to the common search terms, we run separate searches on each of these key conferences checking every paper for relevance.

(This list of 'other' sources grew as a result of applying our search strategy for accepted papers. When completing details about accepted papers, the researcher is prompted to consider secondary searches that are independent of the database search).

Scope: To avoid bias we have selected three bibliographic databases, will include Technical reports, Conference Proceedings and Journal papers. We will follow up secondary studies identified in our primary searches. However, it is beyond the scope of this systematic review to search for and review work in the form of PhD Theses. We therefore exclude PhD theses from our review of the literature on GSD teaching. We also exclude books from our review of the literature.

4 Search Process Documentation

The search process involves two stages. Stage one: Primary search on the 'databases' and 'other sources' listed in 3.2. Stage two: Secondary searches made as a result of identifying work in our primary search.

4.1 Primary search documentation

We document our primary search as follows.

4.1.1 Document: Search terms (tailored for each Database, Journal, Proceeding)

The example below contains search string used in IEEEXplore for RQ1.

Table 1 gives an example of a nested Search String as used in the IEEEXplore database. The Look-up table can be used to check the precise terms used and years included for each recorded paper. We store as much information as possible about each paper in our Summary Spreadsheet and accompanying Endnote file.

Table 1: Search Identifier

IEEEXplore SEARCH TERMS LOOKUP TABLE – 14 June 2015

Researcher Name: Sarah

Date	**Search string** Used Command search and refined by Content Type: Conference Publications Journals & Magazines Year: 2000-2015	Comments IEEEXplore had a limit to number of terms I could use
14 June 2015	**((((software OR "information technology" OR "information system*" OR comput* OR programming) AND (student OR trainee OR learner)) AND ("distributed software" OR "global software") AND (educat* OR train* OR course)))**	Inclusive search: Applies to both RQ1 AND RQ2 – did not limit the papers by including BOOLEAN 'AND' for challenges (RQ1) and recommendations (RQ2).

This yielded 545 papers.

When we develop our search strings for the ACM and Scopus database on our list (in section 3.2), we place them in Appendix B and give them a unique reference. This is necessary as databases tend to have proprietary search methods (e.g. different syntax, nesting allowances, etc). All search strings will be tested to ensure that key texts (known to be in the particular database) are extracted in the search.

Validating selection process of IEEExplore papers.

All 545 IEEEXplore papers were circulated to three key researchers for validation and selection based on title and abstract. All three coded the papers as either Accept; Reject; Not sure; Background. Where there were disagreements, discussions were held. In each instance a 100% agreement was reached without the need for arbitration.

4.1.2 Document: We record the fields in our Data Extraction Form -example of this form is found in Appendix C

4.1.2.1 Document: Study Type
Within our Data Extraction form (in Appendix C), we define the type of study according to Valentine's taxonomy. See Appendix D for the 6 classifications.

4.1.3 Document: Quality Assessment
See Appendix F for the quality assessment scheme. We have not implemented this in the initial version of the SLR, since we use Valentine's taxonomy as a first classification of the rigor behind the method used in the study along with how the study is reported. (See Appendix C)

4.1.4 Document: Accepted papers/Follow-up Form
If a paper passes through our exclusion criteria, meets our inclusion requirements and has been given a quality score, results are abstracted and recorded against the relevant research question(s). This is not a description of the paper, but a list of results. For full description of our exclusion, inclusion criteria see Appendix A.

The accepted papers/follow-up form includes prompts for secondary source follow-up. This form can be used for secondary sources even if the primary paper isn't accepted.

4.1.5 Document: Secondary Search
This is similar to primary search documentation, other than no search string/lookup table will be used. We do not constrain the papers found during this 'snowballing' to be within our date constraints (can pre-date year 2000), may not be present in our IEEEXplore and ACM databases, etc. Our Spreadsheet is used in the same way to record the references as for primary studies. The one exception is that for secondary sources, the 'search string' field in the Spreadsheet is filled in with the details of the primary source that led to this paper being identified along with words "secondary search". We also add the search term, if this is used, e.g. author "Clear". The Field "Name of reference database" is filled in to give information on where search took place, e.g. IEEE *X*plore or ACM.

4.1.6 Document: Procedure for conducting the search
To ensure that the procedure is reliable and replicable, three researchers used this prescriptive process in a pilot study. The outcome of this trial resulted in the following procedural document which we will use for all our primary searches.

Data
Each researcher performing the systematic review will be given the same Data:

Reference Data:
- Our Research Questions
- Exclusion Criteria
- Inclusion Criteria
- Quality Criteria (Valentine's taxonomy) Output Data:
- Generic Results Form.doc
- *For practical purposes all results are combined into one document/excel spreadsheet.*

4.1.7 Document: Specific Guidelines

The information will be stored in google docs folders, one for this **SLR protocol** and its versions, another for the **forms** and summary spreadsheet and a separate folder for the **inclusion-exclusion criteria**. The link to the google drive is given below:

https://drive.google.com/a/aut.ac.nz/folderview?id=0B_tof1dm8dY4fnFQYk1zdXlWMGlSVkpYOGZSd0YyUWNiRElaaTI3RTFiVmdIQXQ4R3VTQ00&usp=sharing_eid

Each paper is given a separate spreadsheet to extract the data, and identified through the unique paper id (issued to each paper when extracted from the database); i.e. IEEE_1; or ACM_1. The mapping of unique no.ID to reference number used in the SLR is given in AppendixG.

4.1.7.1 Completion of Systematic Review

At the end of primary and secondary study data extraction and reporting, we examine the following:

Papers Pending Decision & Papers for Arbitration (to try to progress)
Papers Accepted and Papers Rejected (for notes in case of disagreement)

WIP papers are categorised into the reasons they have not been progressed. A common reason is that a full paper is not readily accessible. Where possible, a decision is made whether to reject or accept. If a decision cannot be reached by the researcher alone, the paper goes to arbitration.

Accepted papers. Each accepted paper will be reviewed by two researchers. Where researchers disagree, the paper goes to external arbitration.

Papers that may go to arbitration fall into the following categories:

(a) Papers that are pending Decision (researchers just don't know)
(b) Papers that have been not been accepted by all researchers

Stage 1: Internal Arbitration: Researchers involved in the data extraction will try to reach an agreement on all papers (whether to include or exclude).

If there is still no agreement, the papers go to stage 2, external arbitration.

Stage 2: External Arbitration: If the internal arbitration fails to reach an agreement then a third independent researcher reviews the paper to make a decision.

4.1.7.2 Multiple Publications/repeated studies

Considering all 'Accepted Papers', searches are made for articles that report the same study. This is done by grouping papers by author (and co-authors). Duplicate work may not be referenced by the author directly therefore papers grouped by author need to be carefully read to uncover possible duplication. Where duplication is found we include only one paper in our review (that we consider to be the best quality – e.g. the most thorough and ideally most up-to-date). Duplicate papers are removed from 'Accepted Papers' list and placed the duplicate papers repository. In this way we avoid giving one finding too much prominence.

4.1.8.1 Document: Data Synthesis Theme Building

Six researchers examined results of data extractions from 10 papers. Taking an inductive approach and through individual ratings, discussion and by consensus we came up with an initial set of themes. Then going forward with an initial set of codes; we took an deductive approach and mapped the new papers to the new codes. Where no code existed for a given recommendation/challenge, a new code was added.

In order to validate the codes, 80 coded snippets were extracted from 6 rich papers (coded by three different researchers). A 7th researcher (who was not part of the code generation exercise) then looked at the themes and mapped each of the 80 code snippets to one of the Major and Minor themes. The validation sheet is given in AppendixE.

4.1.8.2 Document: Duplicate Removal

During the review of papers that made it through to the second round of review, duplicate papers are removed from the pool. Duplicate papers are defined as papers written by the same author, or group of authors, that describe the same experiment, explore/re-hash the topic without going into any significant additional area, or present the same findings in a different publication venue. This is done to ensure that no research group or single experiment/experience is over-represented in the final set of reference papers. Care was taken to ensure that similar papers which contribute in different areas are not identified as duplicates. When a duplicate is identified, the most recent paper, or paper published in an archival outlet (journal) is retained in preference over older papers.

The process by which duplicate papers are identified in the second pass over the the pool of papers is:
1. Complete at least 50% of the reviews in order to get an appreciation of range of papers and topics.
2. Review the list of papers, ordered by author and look at each for similarities based on title and abstract.
3. Discuss the papers with the reviewers if they have been reviewed, and make a decision
4. Check publication dates and venues to identify the most recent version of the paper.
5. Mark older papers as duplicates of the content overlaps in a significant way such that there is no additional contribution in terms of identifying challenges, and opportunities in the field.

4.1.8.3 Document: Data Synthesis

Data synthesis forms will bring together all the findings reported in our Accepted papers/Follow-up forms (Document 4.1.4 in this protocol). The synthesis comprises qualitative lists of findings that will provide broad answers to our research questions. In order to perform sensitivity analysis we categorise the quality, population, location, year and type of study.

There are three forms:
- Data Synthesis Form 1: lists findings of each paper according the research question.
- Data Synthesis Form 2: categorises the findings and notes how many papers agree with each finding.
- Data Synthesis Form 3: Is a sensitivity analysis and separates the findings identified in Data Synthesis Form 2 to see whether there are any differences in the identified groups.

Data Synthesis Form 1: Research Question 1
# of papers accepted that relate to this question (completed at end):

RQ1: What are the key challenges in delivering GSE courses to SE Students?						
Paper ID	Quality (score)	Population (e.g. age group, experience level)	Geographical location(s)	year of study	Type Study	› GSD Education Challenges (list)
Paper ID	Quality score	Population (e.g. age group, experience level)	Geographical location(s)	year of study	Type of Study	GSD Education Challenges (list)
etc						

Data Synthesis Form 1: Research Question 2
# of papers accepted that relate to this question (completed at end):

RQ2: What are the key recommendations for delivering GSE courses to SE Students?						
						Recommendations
Paper ID	Quality	Population	Geographical location	year of study	Type of Study	For GSD education (list)
Paper ID	Quality	Population	Geographical location	year of study	Type of Study	For GSD education (list)
etc						

When findings have been recorded in these summary forms, a finer-grained classification of themes is conducted. We now class synthesis the findings as shown in this example:

Data Synthesis Form 2: Counts of Identified factors

RQ1: What are the key challenges in delivering GSE courses to SE Students?		
GSD education challenge A (identified in Form 1)	# of papers	
GSD education challenge B (identified in Form 1)	# of papers	
etc		

A data synthesis for all RQs will be performed based counts of identified factors reported in Form 1.

When we have identified all the factors we run a sensitivity analysis as shown in example Data Synthesis Form 3:

Data Synthesis Form 3: Sensitivity Analysis based on population for RQ1

RQ1: What are the key challenges in delivering GSE courses to SE Students?			
Population	# of papers	Differences (list)	Similarities (list)
e.g. Students			
e.g. trainers			
e.g. Industry trainees			
e.g. Experienced Practitioners			

Sensitivity analyses (highlighting similarities and differences between groups) will be performed for ALL RQs based on: Population; Geographical Area; Chronology; Study Type (e.g. empirical versus theoretical studies), Data collection method (e.g. questionnaire versus participant observation). When populating the results forms for each individual paper we may find further categories to investigate.

5. Validation of review process

This section explains how we validate our systematic review process - this is in four parts.

The Pilot – Testing the Process

a. Three independent researchers use a subset of resources to test the process. Problems in replicating the process are identified, process is refined accordingly (This stage is completed)
b. Gaps in our searches are identified and search terms and resources are changed to include missing papers.
c. Data Extraction. We test the reliability of how we extract details from accepted papers. An independent researcher, not involved in the pilot, is given a set of accepted papers and asked to fill in the final report.

The review – Testing reliability of selection

d. 100 papers will be reviewed by at least two researchers independently. These represent the first 100 papers extracted from IEEEXplore.

5.2 Testing Boundaries/scope:

The scope of this study is sometimes dictated by limitations of databases (which is beyond our control), or by retaining the focus of our research questions. We found following the guidelines of inclusion/exclusion criteria and quality criteria clear.

5.3 Validation of the Protocol

This first draft is circulated to Tony Clear.

Major amendments to the protocol will be made in accordance with all feedback and reviews. The revised version will underpin the review. Should any further changes be required we will update this protocol and change the version number accordingly. The most up-to-date version of the review will be posted on the WG repository in Google Docs so that all researchers involved in the review have access to the current version.

6. Schedule of Activities

Although the Working group met for only four days, the entire paper writing process took 5 months, starting End May 2015, with leaders planning and writing the protocol, downloading papers for review. The camera ready copy was submitted end October 2015.

Activity	Date	People involved	Completion Date	comments
Planning and Preparation				
Protocol is developed v1	30 May 2015	Sarah	14. June 2015	Completed
Protocol v1 circulated for comment	14 June 2015	Tony and John	20 June	Please let Sarah know if you can't get comments back by this time
Revise accepted papers form	June 25	Sarah, Tony, John	1st July	Based on feedback
Amend protocol and forms	June 28	Sarah	1st July	Based on feedback
Protocol v2 posted on shared repository	June 30	Sarah	1st July	Version used in actual Review
Conduct Review				
Stage 1	14 June 2015	Sarah/ John B		from IEEExplore
Download papers				database
Stage 2 Check Exclusion/inclusion criteria	July (pre wg meeting)	Sarah, Tony and John		Assess papers based on title and abstract (accept, reject, don't know, background).
Stage 3 Check Agreement	July (pre wg meeting)	Sarah (John B and Tony)		Disagreements highlighted. Reviewers discuss
Stage 4 Circulate accepted papers to WG	In Vilnius/ working group	Sarah John B via Googledocs	Completed after WG Sept 20	Two reviewers per paper. (Full papers)
Stage 5 Complete all forms (data extraction)	Conducted at WG in Vilnius	All		1 reviewer per paper
Perform Inter-rater reliability test	Conducted at WG	John Noll		Check agreement levels from extracted themes/coding
Arbitration (2)	Not required			
Synthesise Data	August	JohnB, Sarah, Tony and John N		synthesise data

Publish Results				
Report the review	August/Sept	Sarah		Produce Protocol TR
Report findings	Sept 6	All		Submit SLR for review
Address reviewer commends	11 October	All	30 Oct	Submit camera ready

7. Reporting the review

We plan to publish the process and results of performing the systematic literature review on GSD education in the ITiCSE Working Group Proceedings, which will also be made available through the ACM Digital Library. This will be supported by this detailed technical report that provides all the necessary transparency into the process and final reports.

8. Making changes to the Protocol

It is likely that changes to the protocol will be made when applying the procedures in new situations. Some changes will be made out of necessity, whereas other changes may be made to improve the current process. Every change to the protocol will be recorded and the protocol updated accordingly.

9. References

[1] B. Raza, S. MacDonell, and T. Clear, "Research in Global Software Engineering: A Systematic Snapshot," in *Evaluation of Novel Approaches to Software Engineering*. vol. 417, J. Filipe and L. Maciaszek, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 126-140.

[2] J. Noll, S. Beecham, and I. Richardson, "Global Software Development and Collaboration: Barriers and Solutions " *ACM Inroads*, vol. 1, pp. 66-78, Sept 2010.

[3] E. Conchuir, P. Agerfalk, H. Olsson, and B. Fitzgerald, "Global software development: where are the benefits?," *Commun. ACM*, vol. 52, pp. 127-131, 2009.

[4] I. Richardson, V. Casey, F. McCaffery, J. Burton, and S. Beecham, "A process framework for global software engineering teams," *Information and Software Technology*, vol. 54, pp. 1175-1191, 2012.

[5] M. J. Monasor, A. Vizcaino, M. Piattini, and I. Caballero, "Preparing Students and Engineers for Global Software Development: A Systematic Review," in *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, 2010, pp. 177-186.

[6] M. J. Monasor, A. Vizcaíno, and M. Piattini, "Cultural and linguistic problems in GSD: a simulator to train engineers in these issues," *Journal of Software: Evolution and Process*, vol. 24, pp. 707-717, 2012.[7] T. Clear, F. Young, M. Goldweber, P. Leidig, and K. Scott, "ITiCSE 2001 Working Group Reports - Resources for Instructors of Capstone Courses in Computing," *SIGCSE Bulletin*, vol. 33, pp. 93113, 2001.

[8] T. Clear, G. Claxton, S. Thompson, and S. Fincher, "Cooperative and Work-Integrated Education in Information Technology," in *International Handbook for Cooperative & Work-Integrated Education*, R. Coll and K. Zegwaard, Eds., 2 ed Lowell, MA: World Association for Cooperative Education Inc, 2011, pp. 141-150.

[9] W. Aspray, F. Mayadas, and M. Vardi, "Globalization and Offshoring of Software - A Report of the ACM Job Migration task Force," ACM, New YorkDec 2006.

[10] W. Aspray, A. F. Mayadas, M. Y. Vardi, and S. H. Zweben, "educational response to offshore outsourcing," in *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, ed Houston, Texas, USA: ACM, 2006, pp. 330-331.

[11] M. Daniels, M. Petre, V. Almstrum, L. Asplund, C. Bjorkmann, C. Erickson, *et al.*, "RUNESTONE, an International Student Collaboration Project," in *IEEE Frontiers in Education Conference*, Tempe, Arizona, 1998.

[12] M. Daniels, A. Berglund, and M. Petre, "Reflections on International Projects in Undergraduate CS Education," *Computer Science Education*, vol. 9, pp. 256-267, 1999.

[13] B. Bruegge, A. Dutoit, R. Kobylinski, and G. Teubner, "Transatlantic Project Courses in a University Environment," in *7th Asia-Pacific Software Engineering Conference (APSEC),*, Singapore, 2000.

[14] O. Gotel, V. Kulkarni, M. Say, C. Scharff, and T. Sunetnanta, "Quality Indicators on Global Software Development Projects: Does "Getting to Know You" Really Matter?," in *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, 2009, pp. 3-7.

[15] K. Swigger, R. Brazile, B. Harrington, X. Peng, and F. Apaslan, "Teaching Students How to Work in Global Software Development Environments," presented at the International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006 (CollaborateCom 2006), Atlanta, Georgia, USA, 2006.

[16] T. Clear and D. Kassabova, "A Course in Collaborative Computing: Collaborative Learning and Research with a Global Perspective," in *Proceedings of the 39th ACM Technical Symposium on Computer Science Education*, M. Guzdial and S. Fitzgerald, Eds., ed Portland, Oregon: ACM, 2008, pp. 63-67.

[17] A. Hauer and M. Daniels, "A learning theory perspective on running open ended group projects (OEGPs)," in *Conferences in Research and Practice in Information Technology*. vol. 78, Simon and M. Hamilton, Eds., ed Wollongong, NSW, Australia: ACS, 2008, pp. 85-92.
[18] M. Daniels, Å. Cajander, A. Pears, and T. Clear, "Engineering Education Research in Practice: Evolving Use of Open Ended Group Projects as a Pedagogical Strategy for Developing Skills in Global Collaboration (Special issue on Applications of Engineering Education Research)," *International Journal of Engineering Education* vol. 26, pp. 795-806, 2010.
[19] M. Daniels, Å. Cajander, T. Clear, and R. McDermott, "Collaborative Technologies in Global Engineering: New Competencies and Challenges " *International Journal of Engineering Education* vol. 31, pp. 267-281, 2015.
[20] T. Clear, "Exploring the notion of 'cultural fit' in global virtual collaborations," *ACM Inroads,* vol. 1, pp. 58-65, Sept 2010.
[21] T. Clear, "E-Learning: A Vehicle for Transformation or Trojan Horse for Enterprise? - Revisiting the role of Public Higher Education Institutions," *International Journal on E-Learning,* vol. 1, pp. 15 - 21, October-December 2002.
[22] A. Pears and M. Daniels, "Developing global teamwork skills: The Runestone project," in *Education Engineering (EDUCON), 2010 IEEE*, 2010, pp. 1051-1056.

The end of the Protocol

APPENDIX A:

This appendix defines the scoping of the study as presented through our inclusion and exclusion criteria

INCLUSION CRITERIA

1. Must address global software development/engineering (GSD/GSE) which is defined as collaboration across one or more of three dimensions (global distance): cultural/linguistic, temporal, and geographic.
2. Both theoretical studies and empirical studies
3. Years 2000-date (as in our primary searches in ACM/Scopus/IEEE Xplore); our secondary searches can be any date - no restriction.
4. Must be peer reviewed
5. Must directly answer one or more of our RQs.
6. Must be a primary study

EXCLUSION CRITERIA

1. Books, presentations, opinion pieces, posters, very short papers (less than 2 pages), proposals.
2. Repeated studies (will check this at end - i.e. papers with different title/author order stating the same thing)
3. If focus is primarily on open source development rather than global software development (though open source development is distributed, we want to prepare students for globally software development)
4. Proceedings (references to complete proceedings, not individual papers).
5. E-learning, remote learning, cloud if external to GSD/GSE. (although interested in e-learning tools and virtual learning environments, we focus our research on courses that are in a university setting).
6. Hardware/Distributed systems (where distributed relates to the system, rather than the team).
7. Collaborative software development (if not globally dispersed).
8. No active participation in (at least) parts of the life cycle development process across collaborative groups/parties
9. We exclude SLRs and Tertiary studies (although retain them to support our background). – We do not want to run the risk of duplicating information we find in the primary studies.

(as at 01/09/2015)

APPENDIX B: Search Criteria

Table 1: ACM digital library SEARCH TERMS LOOKUP TABLE – 16 June 2015

Researcher Name: John Barr

Date	Search string Used the query box provided in the Advanced Search option	Comments Due to the constraints of the advanced search option, two queries were performed, one to search abstracts and one to search titles.
16 June 2015	**(Abstract:software or Abstract:programming or Abstract:comput or Abstract:"information technology or information system") and (Abstract:student or Abstract:learner or Abstract:trainee) and (Abstract:"distributed software" or Abstract:"global software") and (Abstract:educat or Abstract:train or Abstract:course)**	Inclusive Abstract search: Applies to both RQ1 AND RQ2 – did not limit the papers by including BOOLEAN 'AND' for challenges (RQ1) and recommendations (RQ2).
16 June 2015	**(Title:software or Title:"inforation technology" or Title:"information system" or Title:comput* or Title:programming) and (Title:student or Title:trainee or Title:learner) and (Title:"distributed software" or Title:"global software")**	Inclusive Title search: Applies to both RQ1 AND RQ2 – did not limit the papers by including BOOLEAN 'AND' for challenges (RQ1) and recommendations (RQ2). Did not include the restrictions that "educat*", "train" or "course" had to be in the title.

Validating selection of ACM papers.

The abstract search yielded 41 papers and the title search yielded 16 papers.

(Data extraction string used for IEEEXplore search is in the body of the Protocol). **Appendix C: Data Extraction Form**

FIELDS TO COMPLETE (PHASE 1)	Your Response	Comments
Paper ID:		Use identifier from master /accepted papers list e.g. IEEE_1 or ACM_1 etc.
Paper Title		First few words will suffice
Researcher Name		Your name
Date researcher analysed this paper:		When you completed this form
EXCLUSION/INCLUSION CRITERIA		
Excl Criteria (a): Is study external to global software engineering?		only interested in GSE/GSD as the focus
Excl Criteria (b): Is study external to teaching and learning?		needs also to be focussed on education
Excl Criteria (c): Is study based on personal opinion/viewpoint?		needs a level of rigour so we can trust the results (even from an expect) - anything without a good theoretical foundation or based on evidence/empirical study we reject
Excl Criteria (d): Is this a repeated study?		include key study only (most comprehensive), repeating results when author publishes in several venues will bias our results
Incl Criteria (a): RQ Answered?		State which RQ is addressed in this study (can be both)
Inclusion Criteria (b): Acceptable source?		Exclude: Books, Book chapters; PhD theses, Tech reports, non-peer reviewed sources, posters, proceeding front matters/sets or short papers (<=two pages). Incl conference/workshop proceedings and journal papers.
DECISION		
Decision Status: {Accept/Reject/Waiting for Full paper/Don't Know}		"Don't know" decisions will go to arbitration. Please use exact wording, as papers will be classified according to how you code this field.
Decision Based on: {Abstract/ Intro/ Conclusion/ Method/Whole Paper/ Peer Review/ Arbitration}		at what point did you make your decision
CONTEXT OF STUDY		

Course / subject taught : (one per row – add more if needed)		Applies to theoretical & empirical studies; e.g. a course on cultural awareness in GSD can be an e-learning training tool or an in-class course.
Population: {HE student/ practitioner/ other}		ADD more rows if you need to - one per type
Type of study: Valentine's taxonomy		Indicate type: Marco Polo, Tools, Experimental, Nifty, Philosophy, John Henry
For empirical studies add:		
Geographical area : (one country per row, add more if needed)		list countries involved in study (i.e. sites used)
Number of sites used :		give number - if not known state' not given' : use numbers e.g. 2, (not two).

***PHASE 2: Qualitative Data Extraction**. Please complete following ONLY if paper is accepted - i.e. has passed all criteria in Phase 1 above*

QUALITATIVE DATA EXTRACTION	Challenge/Solution	**Major Category (based on themes spreadsheet or other inductively derived categories that emerge)**	**Minor Category (based on themes spreadsheet or other inductively derived categories that emerge)**	**Comment** PLEASE NOTE: Your lists of how study answers our RQs will go into our 'Data Synthesis' stage - where we aggregate all our findings across ALL our accepted papers. So please do not interpret what the authors have found, and try to keep your description very short (one or two sentences per challenge/practice at most)
Challenge in Teaching GSD (RQ1)				RQ1: What are the key challenges in delivering GSE courses to SE Students? List as many as you find (create additional rows if needed - one row per challenge)
Challenge in Teaching GSD (RQ1)				Add more rows if needed; use exact text from column A in new column A.
Recommendation for Teaching GSD (RQ2)				RQ2: What are the key recommendations for delivering GSE courses to SE Students? List as many as you find (create additional rows if needed - one row per recommendation)
Recommendation for Teaching GSD (RQ2)				Add more rows if needed; use exact text from column A in new column A.

Methodology (if experiment)(Action Research, Field Study, Descriptive Case Study, Experience Report)		Describe the method used in the study (if appropriate)

Method/Analytical technique (if experiment) {Questionnaire/survey; Face to face interviews; Observation; Focus Groups, prototyping}		Describe the method used in the study (if appropriate)
Quality of execution (if experimental in line)		
Goal of paper (optional)		What was the overall goal of the study?
Emerging Theme (optional)		List any themes in terms of GSD challenges or recommendations
ADDITIONAL DATA/FOLLOW UP		
Other observations or useful quotes found in paper		Record useful text here / exact quotes we can use in our report
Other observations or useful quotes found in paper		
References found in paper/snowballing (to follow up)		Can pre-date year 2000
References found in paper/snowballing (to follow up)		

Appendix D: Study Type (according to Valentine's Taxonomy, 2004)

Applied to study type field (In Data Extraction Form Appx C).

A six-fold taxonomy to classify the type of articles found in Educational Research. Valentine suggests that we do not need a strictly quantified, statistical model to prove significant educational results. As a result he set "as inclusive (and yet reasonable) a bar as possible for this category" and settled on a simple rubric:

"Experimental": If the author made any attempt at assessing the "treatment" with some scientific analysis. For a minimal example, Bagert, et. al. [1995] showed that after a New Breadth-First CS1 course, the number of CS majors earning a 'C' or better in CS2 doubled at Texas Tech. At the other end of the category, Dey & Mand [1986] did a complete statistical analysis of 500 introductory students at two institutions to show the impact of math background and prior programming to success in CS1. Another, less quantitative example is Fleury [1991] who, through a series of interviews, developed ethnography of how students develop their own (often faulty) cognitive rules about parameter passing. Clancy & Linn [1999] in a philosophical discussion of **pedagogy did a review of existing research literature**, so they were also included here. Please note that this was a preemptive category, so if the presentation fit here and somewhere else (e.g. a quantified assessment of some new Tool), it was placed here.

"Marco Polo": "I went there and I saw this." SIGCSE veterans recognize this as a staple at the Symposium. Colleagues describe how their institution has tried a new curriculum, adopted a new language or put up a new course. The reasoning is defined, the component parts are explained, and then (and this is the giveaway for this category) a conclusion is drawn like "Overall, I believe the [topic] has been a big success." or "Students seemed to really enjoy the new [topic]". Now, Marco Polo presentation serve an important function: we are a community of educators and sharing our successes (and failures) enriches the whole community.

"Philosophy": where the author has made an attempt to generate debate of an issue. E.g. Reed, et. al [2002] who who discussed "Integrating Empirical Methods into CS", and said, "This panel is designed to promote discussion …within the traditional computer science community." Or McCraken [1992] who tried to stimulate the core language debate along philosophical and educational lines. Of course the "Denning Report" [1988] on "Computing as a Discipline" was a foundational work that still guides our philosophical understanding.

"Tools": Among many other things, colleagues have developed software to animate algorithms, to help grade student programs, to teach recursion, and to provide introductory development platforms. For example, Studer et. al [1995] developed a tool so novice programmers could use pictograms rather than syntax to create programs. Rambally [1985] built a tool to graphically represent linked data structures for students. Not all tools were software; an author could present a paradigm or an organizing rubric to be a tool for an entire course. Carrasquel et. al. [1989] presented a combination of a visual design tree and data flow diagrams as an effective teaching tool for CS1.

"Nifty". Nifty assignments, projects, puzzles, games and paradigms are the bubbles in the champagne of SIGCSE. Most of us seem to appreciate innovative, interesting ways to teach students our abstract concepts. Sometimes the difference between Nifty and Tools was fuzzy, but generally a Tool would be used over the course of a semester, and a Nifty assignment was more limited in duration. Ginat [1995] related loop invariants to mathematical games. Fell and Proulx [1997] showed how to use Martian planetary images in CS1. Cigas [1992], in a real gem, shows how to use finite state automata in traditional CS1/CS2 problems to improve student success.

"John Henry": .. a course that seems so outrageously difficult (in my opinion), that one suspects it is telling us more about the author than it is about the pedagogy of the class. E.g., … you could teach CS1 as a predicate logic course in IBM 360 assembler – but why would you want to do that? Yes, every once in a while somebody can beat the steam engine, but most of us try to avoid that.. John Henry's are valuable to our community, too. We should continually be touching that upper limit of our pedagogy (which means occasionally we'll push over the line).

Source: Valentine, D. W. (2004). *CS educational research: a meta-analysis of SIGCSE technical symposium proceedings*. presented at the meeting of the Proceedings of the 35th SIGCSE technical symposium on Computer science education, Norfolk, Virginia, USA. doi:10.1145/971300.971391

Appendix E: Code Validation sheet (Taken from Excel Spreadsheet)

Code ID	paper Id	Major Category *Completed later*	Minor Category *Completed later*	Challenge/ Recommendation	Detail from paper	Original Coder
1	IEEE_49			Challenge:	Managing customers and the development process. Customers wanted additional functionality etc.	MO
2	IEEE_49			Challenge:	Dealing with problems at the customer site that impacts on student progress and may cause work redistribution within the team.	MO
3	ICGSE_2			Challenge:	lingua franca as a second language	MO
4	ICGSE_2			Challenge:	One group was not willing to ask questions of instructors while the students at the other location were comfortable doing so. One location had students who were more independent thinkers and had better managerial skills difference in previous educational background) then the other location.	MO
5	ICGSE_2			Challenge:	High grades were not celebrated as much in one of the locations and this impacted the performance of the students at that location.	MO
6	IEEE_19			Challenge:	Cultural differences - The US and Cambodia have different culture, different educational systems and students made assumptions based on their own experiences which were not applicable to the other site. The work ethic also differed at the two locations. US students had to learn to compromise as they assumed that the Cambodian students would fit in with them.	MO
7	IEEE_19			Challenge:	time zones - a 12 hour time differnce between locations made it difficult for the students to coordinate activities and "meet" regularly. Students tended to prefer asynchronous communication.	MO
8	IEEE_19			Challenge:	scope creep - unlike projects that students create for themselves, the US students were developing code for the Cambodian students (clients) and scope creep was a concern.	MO
9	IEEE_19			Challenge:	negotiation and accountability - unlike projects suggested by students where there is no third party client, students had to produce deliverables. They typically rose to the occasion and provided a high degree of professionalism.	MO
10	IEEE_19			Challenge:	communication - In addition to time zone differences, English language was a challenge for the Cambodian students and this lead to some communication failures.	MO

11	IEEE_19	Challenge:	team leadership - strong team leadership was necessary for success.	MO
12	IEEE_19	Challenge:	just-in-time learning - teaching content as the students need it made it difficult for students to plan far enough into the future.	MO
13	IEEE_66	Challenge:	students don't start communications	JB
14	IEEE_66	Challenge:	students lack loyalty, team spirit and collective responsibility	JB
15	IEEE_66	Challenge:	risk that communication decreases.	JB
16	IEEE_66	Challenge:	forget the other (global) team	JB
17	IEEE_66	Challenge:	students with different backgrounds have different sources of motivation	JB
18	IEEE_66	Challenge:	language differences causes difficulties understanding other site	JB
19	IEEE_66	Challenge:	technical capabilities differ between students at different sites and within teams on same site. Causes problems in coordinating development	JB
20	IEEE_66	Challenge:	some students tend to be more open and direct in their conversation, some are more reserved in giving their opinions and avoid confrontation	JB
21	IEEE_66	Challenge:	some students had more flexible interpretation of time	JB
22	IEEE_66	Challenge:	commitment	JB
23	IEEE_66	Challenge:	different understandings of teamwork	JB
24	IEEE_66	Challenge:	tolerance of diversity	JB
25	IEEE_66	Challenge:	tolerance of difference	JB
26	IEEE_66	Challenge:	combination of two inflexible sets of rules from different institutions brings unsolvable situations and a lot of headaches due to inefficiency in many procedures.	JB
27	IEEE_48	Challenge:	need to mentor students	JB
28	IEEE_48	Challenge:	need to audit student work	JB
29	IEEE_48	Challenge:	one team felt not included, caused competition	JB

30	IEEE_48	Challenge:	Need to plan …	JB
31	IEEE_48	Challenge:	Monitor and be alert	JB
32	IEEE_48	Challenge:	Reflect and close project	JB
33	ACM_14	Challenge:	coupling of the participating teams'	JN
34	ACM_14	Challenge:	how to handle risks and failures'	JN
35	ACM_14	Challenge:	Integration failures before deadlines	JN
36	ACM_14	Challenge:	Integration failures before deadlines	JN
37	IEEE_49	Recommendation:	Daily meetings - teams need to meet briefly each day in order to stay focussed and coordinated.	MO
38	IEEE_49	Recommendation:	Group leader has additional responsibility to coordinate and manage a dispersed team.	MO
39	IEEE_49	Recommendation:	Version control is more important in a distributed context.	MO
40	IEEE_49	Recommendation:	Balance the expertise within each group so that each group has a range of skills available to it.	MO
41	IEEE_49	Recommendation:	Pair programming on-line works as long as it is supported by E-mail, chat sessions, and instant messaging.	MO
42	IEEE_49	Recommendation:	Have the teams involved in training other team members to practice skills transfer and help balance workload.	MO
43	ICGSE_2	Recommendation:	The authors recommend identifying the cultural and educational differences between the students in the 2 locations and then exploit those differences through knowledge transfer in the delivery of the course.	MO
44	ICGSE_2	Recommendation:	The authors recommend identifying the cultural and educational differences between the students in the 2 locations and then exploit those differences through knowledge transfer in the delivery of the course.	MO
45	IEEE_19	Recommendation:	Set-up and Managerial costs - the scope of the project and the stucture to be used was determined by agreement betwen the instructors at the 2 sites. The students missed out on having that experience.	MO
46	IEEE_19	Recommendation:	independent oversight - it would be helpful to have an independent faculty member have some oversight to keep the bigger picturein mind. It is too easy for the instructor to become focused in minute details.	MO
47	IEEE_19	Recommendation:	Just-in-time learning - this did not allow the students to have a full understanding of the whole process and be able to apply it. It would be better if they had a software engineering class first to learn the skills and then	MO

			be able to apply them in a global context.	
48	IEEE_66	Recommendation:	Start communication by brute force	JB
49	IEEE_66	Recommendation:	Get the students to be familiar with each other as soon as possible	JB
50	IEEE_66	Recommendation:	Keep communication levels consistently high	JB
51	IEEE_66	Recommendation:	Ensure that students keep the other site in mind	JB
52	IEEE_66	Recommendation:	Keep the students highly motivated	JB
53	IEEE_66	Recommendation:	a. Give students enough flexibility to develop their creativity	JB
54	IEEE_66	Recommendation:	Give students the opportunity to express themselves through the presentations	JB
55	IEEE_66	Recommendation:	c. use a videoconference system	JB
56	IEEE_66	Recommendation:	d. Awards and positive competition	JB
57	IEEE_66	Recommendation:	Remember: we are different a. lecture about cultural differences and students are given an assignment to compare different cultures.	JB
58	IEEE_66	Recommendation:	Remember: we are different b. close supervision of teams	JB
59	IEEE_66	Recommendation:	Be flexible – overcome the differences a. place students into project groups with care and insight	JB
60	IEEE_66	Recommendation:	Be flexible – overcome the differences b. select the project technology, project requirements and goals based on the students' experience	JB
61	IEEE_66	Recommendation:	Be flexible – beat the administration a. absolute flexibility and creativity of the teaching staff in finding solutions, and a full understanding of the constraints faced by the other site	JB
62	IEEE_66	Recommendation:	Be alert new problems can arise at any time	JB
63	IEEE_66	Recommendation:	Be enthusiastic: teaching staff must be enthusiastic above and beyond the standard level.	JB
64	ACM_14	Recommendation:	keep project simple	JN

65	ACM_14	Recommendation:	allocate different modules of large system to distributed teams	JN
66	ACM_14	Recommendation:	use design by contract (in Eiffel) to specify module/subsystem interfaces	JN
67	ACM_14	Recommendation:	require designated group project manager	JN
68	ACM_14	Recommendation:	require designated institution project manager	JN
69	ACM_14	Recommendation:	provide report document templates	JN
70	ACM_14	Recommendation:	require mandatory code review for API	JN
71	ACM_14	Recommendation:	require mandatory project communication plan	JN
72	ACM_14	Recommendation:	require mandatory project communication plan	JN
73	ACM_14	Recommendation:	give students the choice of co-located or distributed project	JN
74	ACM_14	Recommendation:	hold pre-semester training sessions	JN
75	ACM_14	Recommendation:	have optional group exercises emphasising communication skills	JN
76	ACM_14	Recommendation:	have optional group exercises emphasising management skills	JN

Please select one of the codes on "themes-challenges" that you think best maps to each recommendation/challenge. If you cannot find a suitable code please state "none found to fit"; if you just don't know (might need more context), please state "don't know". But please try to map the text to (a) a major theme, and (b) a minor theme if you can!

If you want to check full papers go to googledocs:
https://drive.google.com/drive/folders/0B6yUKMYGAI6Cfkk1eWZGTGpScDVsNkpjMmRIWGk2TFhZX1o5SmdfcHJlcXBfSzJlVHFuRFU

Appendix F: Quality Assessment Scheme (proposed)

We planned to complete a quality assessment for ALL papers that have passed the exclusion and inclusion assessments. The quality assessment form lists and aggregates quality criteria. The objective is to provide a rough guide to the quality of the paper before completing the accepted papers form. This assessment does not act as an exclusion criterion but guides interpretation. The score alone has little meaning; to understand the quality we need to look at the criteria and context of the assessment and cannot compare quality of different papers as based on the score alone. – We plan to conduct the quality assessment at a later date.

Table F1: QUALITY ASSESSMENT

Item	**Assessment criteria**	**Score between 0 – 1**	**Response options for Score**
1 Aims of the Research	Is there a clear statement of the aims of the research? *Does the study present empirical data or theoretical hypothesis?* *Is there a clear, unambiguous statement of the study's primary outcome based on evidence & argument?*		Yes = 1 /No = 0
For empirical studies:			
2 Context description	Is there an adequate description of the context in which the research was carried out? *Study type? Number of sites, Course taught, Course Level, Countries involved, Length of course, Type of student.*		Yes = 1 /No = 0
3 Sampling	Was the recruitment strategy appropriate to the aims of the research? *Were the cases representative of our defined population? (How typical is this population?*		Yes = 1/ No = 0
4 Data Collection	Were the data collected in a way that addressed the research issue? *Is it clear how the data were collected?* *Has the researcher justified the methods chosen?* *How rigorous was the method (go to next table (F1.1) for breakdown of*		Yes = 1/ No = 0

	scores.		
5 Data Analysis	Was the data analysis sufficiently rigorous? *Was there an in-depth description of the analysis process?* *Has sufficient data been presented to support the findings?*		Yes = 1/ No = 0
6 Reflexivity	Has the relationship between researcher and participants been adequately considered? *Did the researchers critically examine their own role, potential bias, and influence during: research question formulation, sample recruitment, data collection, and analysis and selection of data for presentation?*		Yes = 1/ No = 0
7 Findings	Is there a clear statement of the findings? *Are the findings explicit (e.g. magnitude of effect)? Are the limitations of the study discussed explicitly?*		Yes = 1/ No = 0
For theoretical studies:			
8 References	Is the paper well/appropriately referenced? *Can the reader trace where the recommendations/challenges came from?*		Yes = 1 Moderately = .5 No = 0
9	Are the recommendations/challenges based on previous research (i.e. the paper has a good background section to show how the recommendations / framework/ model came from).		Yes = 1 Moderately = .5 No = 0
10	Could the reader replicate the process?		Yes = 1 No = 0
11	Has the model/framework/set of recommendations/challenges been validated?		Yes = 1 No = 0
*Total Quality Score			*Enter this score in the data extraction form in Quality*

		assessment field

Table F1.1: Coding and Scoring Data collections		
Data collection Method	****Code**	**Score (Sample No)**
Questionnaire/Survey (self completed)	1	Unit = 1 person <=5 = 0; >5<50 =.5; >50 = 1
Face to face interviews	2	Unit = 1 person Depends on depth of interview. Heuristic <3 = 0; ≥3 ≤5 = .5; >5 = 1
Observation	3	Unit = 1 person Depends on depth and time spent. Heuristic <3 = 0; ≥3 ≤5 = .5; >5 = 1
Focus Groups	4	Unit = Group Depends on depth and time spent. Heuristic <3 = 0; ≥3 ≤5 = .5; >5 = 1
Theoretical Study (no data collection)	5	n/a
Secondary Data used (e.g. systematic literature review)	6	n/a
For empirical studies, enter code number into Spreadsheet/Endnote "Type of Empirical Study" field		
If method not included in this table, Add new row and number here and update protocol accordingly – creating a new version number.		

*Fill in Spreadsheet Field 'Quality Assessment (score)' with Total Quality Score,

**If study is empirical, fill in Spreadsheet Field "Type of Empirical Study" with type of study code given in Table 2.1

APPENDIX G:

References from SLR. Set of 82 accepted papers (with hard-coded numbering and Paper ID mapped)

[#1-82]

[1]: [IEEE_299] Almeida, E., et al. *Teaching Globally Distributed Software Development: An Experience Report*. in *IEEE 25th Conference on Software Engineering Education and Training (CSEE&T)*. 2012. 17-19 April 2012. p. 105-109.

[2]: [ACM_12] Berkling, K., M. Geisser, T. Hildenbrand, and F. Rothlauf, *Offshore software development: transferring research findings into the classroom*, in *Software Engineering Approaches for Offshore and Outsourced Development* 2007, Springer. p. 1-18.

[3]: [IEEE_74] Bosnić, I., I. Čavrak, M. Orlić, and M. Žagar. *Picking the right project: Assigning student teams in a GSD course*. in *IEEE 26th Conference on Software Engineering Education and Training (CSEE&T)*. 2013. 19-21 May 2013. p. 149-158.

[4]: [ACM_3] Bosnić, I., I. Čavrak, M. Orlić, M. Žagar, and I. Crnković. *Avoiding scylla and charybdis in distributed software development course*. in *Proceedings of the 2011 community building workshop on Collaborative teaching of globally distributed software development*. 2011. ACM. p. 26-30.

[5]: [ACM_9] Bosnić, I., I. Čavrak, M. Orlić, M. Žagar, and I. Crnković. *Student motivation in distributed software development projects*. in *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*. 2011. ACM. p. 31-35.

[6]: [IEEE_71] Bosnić, I., I. Čavrak, M. Žagar, R. Land, and I. Crnković. *Customers' Role in Teaching Distributed Software Development*. in *23rd IEEE Conference on Software Engineering Education and Training (CSEE&T)*. 2010. 9-12 March 2010. p. 73-80.

[7]: [CTGDSD_9] Bosnić, I., F. Ciccozzit, I. Čavrak, R. Mirandola, and M. Orlić. *Multi-dimensional assessment of risks in a distributed software development course*. in *Collaborative Teaching of Globally Distributed Software Development (CTGDSD), 2013 3rd International Workshop on*. 2013. IEEE. p. 6-10.

[8]: [IEEE_7] Braun, A., A.H. Dutoit, A.G. Harrer, and B. Brugge. *IBistro: a learning environment for knowledge construction in distributed software engineering courses*. in *Ninth Asia-Pacific Software Engineering Conference*. 2002. 2002. p. 197-203.

[9]: [Scopus_34] Brazile, R.P., K. Swigger, M.R. Hoyt, B. Lee, and B. Nelson. *A System to Support Teaching Global Software Development*. in *American Society for Engineering Education*. 2012. American Society for Engineering Education.

[10]: [IEEE_189] Brooks, I. and K. Swigger. *The role of leadership and its effect on the temporal patterns of global software development teams*. in *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. 2012. 14-17 Oct. 2012. p. 381-390.

[11]: [IEEE_156] Bruegge, B., A.H. Dutoit, R. Kobylinski, and G. Teubner. *Transatlantic project courses in a university environment*. in *Seventh Asia-Pacific Software Engineering Conference, 2000. APSEC*. 2000. 2000. p. 30-37.

[12]: [Scopus_25] Cao, L., H. Zhu, and G. Su. *Global Software Development Project*. in *18th Americas Conference on Information Systems, AMCIS 2012 Proceedings. Paper 15* 2012. July 29.

[13]: [IEEE_164] Carlson, P. and X. Nan. *Experience and recommendations for distributed software development*. in *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD)*. 2012. 9-9 June 2012. p. 21-24.

[14]: [Scopus_84] Casey, V., *Imparting the importance of culture to global software development*. ACM inroads, 2010. **1**(3): p. 51-57.

[15]: [IEEE_54] Cavrak, I., M. Orlic, and I. Crnkovic. *Collaboration patterns in distributed software development projects*. in *34th International Conference on Software Engineering (ICSE)* 2012. 2-9 June 2012. p. 1235-1244.

[16]: [IEEE_234] Clear, T. *Replicating an'Onshore'Capstone Computing Project in a'Farshore'Setting--An Experience Report*. in *6th IEEE International Conference on Global Software Engineering (ICGSE)* 2011. IEEE. p. 161-165.

[17]: [IEEE_258] Clear, T. and M. Daniels. *Using groupware for international collaborative learning*. in *30th Annual Frontiers in Education Conference, 2000. FIE*. 2000. 2000. p. F1C/18-F1C/23 vol.1.

[18]: [Scopus_113] Clear, T., J. Whalley, J. Hill, Y. Liu, A. Pears, and B. Plimmer. *A global software project: Developing a tablet pc capture platform for explanograms*. in *Proceedings of the 8th International Conference on Computing Education Research*. 2008. ACM. p. 41-50.

[19]: [IEEE_66] Crnković, I., I. Bosnić, and M. Žagar. *Ten tips to succeed in global software engineering education*. in *Proceedings of the 34th International Conference on Software Engineering (ICSE)*. 2012. IEEE Press. p. 1225-1234.

[20]: [IEEE_69] Dastidar, S.G. and S. Chatterjee. *Distributed software development: Experience and recommendation*. in *3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)*. 2013. 25-25 May 2013. p. 11-14.

[21]: [IEEE_140] Deiters, C., et al. *GloSE-Lab: Teaching Global Software Engineering*. in *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on*. 2011. 15-18 Aug. 2011. p. 156-160.

[22]: [IEEE_212] Doerry, E., R. Klempous, J. Nikodem, and W. Paetzold. *Virtual student exchange: lessons learned in virtual international teaming in interdisciplinary design education*. in *Proceedings of the Fifth International Conference on Information Technology Based Higher Education and Training, 2004. ITHET*. 2004. 31 May-2 June 2004. p. 650-655.

[23]: [IEEE_172] Ende, M., R. Lammermann, P. Brockmann, and G. Ayurzana. *A virtual, global classroom to teach global software engineering: A Mongolian-German team-teaching project*. in *Second International Conference on e-Learning and e-Technologies in Education (ICEEE)*. 2013. 23-25 Sept. 2013. p. 229-233.

[24]: [IEEE_293] Estler, H.C., M. Nordio, C.A. Furia, and B. Meyer. *Awareness and Merge Conflicts in Distributed Software Development*. in *IEEE 9th International Conference on Global Software Engineering (ICGSE)*. 2014. 18-21 Aug. 2014. p. 26-35.

[25]: [IEEE_152] Fagerholm, F., P. Johnson, A. Sanchez Guinea, J. Borenstein, and J. Munch. *Onboarding in Open Source Software Projects: A Preliminary Analysis*. in *IEEE 8th International Conference on Global Software Engineering Workshops (ICGSEW)*. 2013. 26-26 Aug. 2013. p. 5-10.

[26]: [IEEE_139] Fagerholm, F., N. Oza, and J. Munch. *A platform for teaching applied distributed software development: The ongoing journey of the Helsinki software factory*. in *3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)*. 2013. 25-25 May 2013. p. 1-5.

[27]: [IEEE_9] Farley, A., S. Faulk, V. Lo, A. Proskurowski, and M. Young. *Intensive international Summer Schools in Global Distributed Software Development*. in *Frontiers in Education Conference (FIE)*. 2012. 3-6 Oct. 2012. p. 1-6.

[28]: [IEEE_15] Favela, J. and F. Pena-Mora, *An experience in collaborative software engineering education*. IEEE Software, 2001. **18**(2): p. 47-53.

[29]: [IEEE_159] Feljan, J., I. Bosnić, I. Bosnic, M. Orlić, and M. Žagar. *Distributed Software Development course: Students' and teachers' perspectives*. in *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD)*. 2012. 9-9 June 2012. p. 16-20.

[30]: [IEEE_17] Filipovikj, P., J. Feljan, and I. Crnković. *Ten tips to succeed in global software engineering education: What do the students say?* in *3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)*. 2013. 25-25 May 2013. p. 20-24.

[31]: [Scopus_57] Giraldo, F., S.F. Ochoa, L. Aballay, C. Clunie, A. Neyem, and R. Anaya, *Supporting Instructional Software Engineering Activities Using CODILA: Some Latin American Experiences*, in *Education and Educational Technology*2012, Springer. p. 591-598.

[32]: [IEEE_165] Giraldo, F.D., et al. *Applying a distributed CSCL activity for teaching software architecture*. in *2011 International Conference on Information Society (i-Society)*. 2011. 27-29 June 2011. p. 208-214.

[33]: [ACM_11] Gloor, P., M. Paasivaara, C. Lassenius, D. Schoder, K. Fischbach, and C. Miller, *Teaching a global project course: experiences and lessons learned*, in *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*2011, ACM: Waikiki, Honolulu, HI, USA. p. 1-5.

[34]: [IEEE_48] Gotel, O., V. Kulkarni, M. Say, C. Scharff, and T. Sunetnanta. *A Global and Competition-Based Model for Fostering Technical and Soft Skills in Software Engineering*

Education. in *22nd Conference on Software Engineering Education and Training (CSEET)*. 2009. 17-20 Feb. 2009. p. 271-278.
[35]: [IEEE_291] Gotel, O., V. Kulkarni, M. Say, C. Scharff, and T. Sunetnanta. *Quality Indicators on Global Software Development Projects: Does "Getting to Know You" Really Matter?* in *Fourth IEEE International Conference on Global Software Engineering (ICGSE)* 2009. 13-16 July 2009. p. 3-7.
[36]: [Scopus_108] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Students as partners and students as mentors: an educational model for quality assurance in global software development*, in *Software Engineering Approaches for Offshore and Outsourced Development* 2009, Springer. p. 90-106.
[37]: [IEEE_80] Gotel, O., C. Scharff, and V. Kulkarni, *Mixing continents, competences and roles: Five years of lessons for software engineering education.* IET Software, 2012. **6**(3): p. 199-213.
[38]: [IEEE_19] Gotel, O., C. Scharff, and S. Seng. *Preparing Computer Science Students for Global Software Development*. in *36th Annual Frontiers in Education Conference*. 2006. 27-31 Oct. 2006. p. 9-14.
[39]: [ITICSE_8] Honig, W.L. and T. Prasad. *A classroom outsourcing experience for software engineering learning.* in *ACM SIGCSE Bulletin*. 2007. ACM. p. 181-185.
[40]: [IEEE_284] Inkeri Verkamo, A., J. Taina, Y. Bogoyavlenskiy, D. Korzun, and T. Tuohiniemi. *Distributed Cross-Cultural Student Software Project: A Case Study*. in *18th Conference on Software Engineering Education & Training*. 2005. 18-20 April 2005. p. 207-214.
[41]: [IEEE_52] Junhua, D. *A framework for global collaboration in teaching software engineering*. in *3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)*. 2013. 25-25 May 2013. p. 30-34.
[42]: [IEEE_53] Junhua, D. and Y. Biwu. *Teaching software engineering with Global Understanding*. in *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD)*. 2012. 9-9 June 2012. p. 11-15.
[43]: [Scopus_66] Keenan, E. and A. Steele. *Developing a pedagogical infrastructure for teaching globally distributed software development*. in *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*. 2011. ACM. p. 6-10.
[44]: [Scopus_80] Keenan, E., A. Steele, and X. Jia. *Simulating Global Software Development in a Course Environment*. in *5th IEEE International Conference on Global Software Engineering (ICGSE)*. 2010. IEEE. p. 201-205.
[45]: [ICGSE_2] Lago, P., H. Muccini, and M.A. Babar. *Developing a Course on Designing Software in Globally Distributed Teams*. in *IEEE International Conference on Global Software Engineering (ICGSE)* 2008. 17-20 Aug. 2008. p. 249-253.
[46]: [Scopus_31] Lago, P., H. Muccini, and M.A. Babar, *An empirical study of learning by osmosis in global software engineering*. Journal of Software: Evolution and Process, 2012. **24**(6): p. 693-706.
[47]: [IEEE_58] Lago, P., H. Muccini, L. Beus-Dukic, I. Crnkovic, S. Punnekkat, and H. Van Vliet. *Towards a European Master Programme on Global Software Engineering*. in *20th Conference on Software Engineering Education & Training (CSEET '07)*. 2007. 3-5 July 2007. p. 184-194.
[48]: [IEEE_192] Last, M.Z. *Understanding the group development process in global software teams*. in *33rd Annual Frontiers in Education (FIE)*. 2003. 5-8 Nov. 2003. p. S1F-20-5 vol.3.
[49]: [IEEE_286] Lescher, C., L. Yang, and B. Bruegge. *Teaching Global Software Engineering: Interactive Exercises for the Classroom*. in *IEEE 9th International Conference on Global Software Engineering (ICGSE)*. 2014. 18-21 Aug. 2014. p. 163-172.
[50]: [Scopus_85] Long, J. *Outsourcing in Next Generation Software Engineering Technology Education*. in *American Society for Engineering Education*. 2010. American Society for Engineering Education.
[51]: [IEEE_83] Mäkiö, J. and S. Betz. *On educating globally distributed software development—A case study*. in *24th International Symposium on Computer and Information Sciences (ISCIS)*. 2009. IEEE. p. 480-485.
[52]: [Scopus_59] Matthes, F., et al. *Teaching Global Software Engineering and International Project Management-Experiences and Lessons Learned from Four Academic Projects*. in *CSEDU (2)*. 2011. p. 5-15.
[53]: [IEEE_62] McDermott, R., J. Bass, and J. Lalchandani. *The learner experience of student-led international group project work in software engineering*. in *IEEE Frontiers in Education Conference (FIE)*. 2014. 22-25 Oct. 2014. p. 1-8.
[54]: [IEEE_2] Monasor, M.J., A. Vizcaino, and M. Piattini. *VENTURE: Towards a framework for simulating GSD in educational environments*. in *Fifth International Conference on Research Challenges in Information Science (RCIS)*. 2011. 19-21 May 2011. p. 1-10.

[55]: [Scopus_81] Monasor, M.J., A. Vizcaíno, and M. Piattini, *A tool for training students and engineers in global software development practices*, in *Collaboration and Technology*2010, Springer. p. 169-184.
[56]: [ACM_13] Monasor, M.J., A. Vizcaíno, and M. Piattini, *Providing training in GSD by using a virtual environment*, in *Product-Focused Software Process Improvement (Profes)*2012, Springer. p. 203-217.
[57]: [Scopus_4] Monasor, M.J., A. Vizcaíno, M. Piattini, J. Noll, and S. Beecham. *Assessment process for a simulation-based training environment in global software development*. in *Proceedings of the 2014 conference on Innovation & Technology in Computer Science Education (ITiCSE)*. 2014. ACM. p. 231-236.
[58]: [IEEE_49] Neto, C.R.L. and E.S. De Almeida. *Five years of lessons learned from the software engineering course: adapting best practices for distributed software development*. in *Proceedings of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development*. 2012. IEEE Press. p. 6-10.
[59]: [IEEE_136] Noll, J., A. Butterfield, K. Farrell, T. Mason, M. McGuire, and R. McKinley. *GSD Sim: A Global Software Development Game*. in *IEEE International Conference on Global Software Engineeering Workshops (ICGSEW)*. 2014. 18-18 Aug. 2014. p. 15-20.
[60]: [IEEE_55] Nordio, M., H.C. Estler, B. Meyer, N. Aguirre, R. Prikladnicki, E. Di Nitto, and A. Savidis. *An experiment on teaching coordination in a globally distributed software engineering class*. in *IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*. 2014. 23-25 April 2014. p. 109-118.
[61]: [IEEE_305] Nordio, M., H.C. Estler, B. Meyer, J. Tschannen, C. Ghezzi, and E. di Nitto. *How Do Distribution and Time Zones Affect Software Development? A Case Study on Communication*. in *6th IEEE International Conference on Global Software Engineering (ICGSE)*. 2011. 15-18 Aug. 2011. p. 176-184.
[62]: [ACM_14] Nordio, M., et al. *Teaching software engineering using globally distributed projects: the DOSE course*. in *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*. 2011. ACM. p. 36-40.
[63]: [IEEE_141] Nordio, M., R. Mitin, and B. Meyer. *Advanced hands-on training for distributed and outsourced software engineering*. in *ACM/IEEE 32nd International Conference on Software Engineering*. 2010. 2-8 May 2010. p. 555-558.
[64]: [IEEE_169] Paasivaara, M., C. Lassenius, D. Damian, P. Raty, and A. Schroter. *Teaching students global software engineering skills using distributed Scrum*. in *35th International Conference on Software Engineering (ICSE)*. 2013. 18-26 May 2013. p. 1128-1137.
[65]: [Scopus_158] Peña-Mora, F., R. Struminger, J. Favela, and R. Losey. *Supporting a Project-Based, Collaborative, Distance Learning Lab*. in *Computing in Civil and Building Engineering*. 2000. ASCE. p. 170-176.
[66]: [ITICSE_7] Petkovic, D., G.D. Thompson, and R. Todtenhoefer, *Assessment and comparison of local and global SW engineering practices in a classroom setting*. ACM SIGCSE Bulletin, 2008. **40**(3): p. 78-82.
[67]: [IEEE_16] Petkovic, D., R. Todtenhoefer, and G. Thompson. *Teaching Practical Software Engineering and Global Software Engineering: Case Study and Recommendations*. in *36th Annual Frontiers in Education Conference*. 2006. 27-31 Oct. 2006. p. 19-24.
[68]: [Scopus_146] Richardson, I., A.E. Milewski, N. Mullick, and P. Keil. *Distributed development: an education perspective on the global studio project*. in *Proceedings of the 28th international conference on software engineering*. 2006. ACM. p. 679-684.
[69]: [IEEE_174] Richardson, I., S. Moore, D. Paulish, V. Casey, and D. Zage. *Globalizing Software Development in the Local Classroom*. in *20th Conference on Software Engineering Education & Training, CSEET '07*. 2007. 3-5 July 2007. p. 64-71.
[70]: [IEEE_323] Romero, M., A. Vizcaino, and M. Piattini. *Teaching Requirements Elicitation within the Context of Global Software Development*. in *Mexican International Conference on Computer Science (ENC)*. 2009. 21-25 Sept. 2009. p. 232-239.
[71]: [Scopus_111] Romero, M., A. Vizcaíno, and M. Piattini. *Developing the Skills Needed for Requirement Elicitation in Global Software Development*. in *ICEIS (1)*. 2008. p. 393-396.
[72]: [IEEE_308] Serce, F.C., F.N. Alpaslan, K. Swigger, R. Brazile, G. Dafoulas, and V. Lopez. *Strategies and guidelines for building effective distributed learning teams in higher education*. in *9th International Conference on Information Technology Based Higher Education and Training (ITHET)*. 2010. April 29 2010-May 1 2010. p. 247-253.

[73]: [IEEE_138] Shata, O. *A Crash Undergraduate Course in Global Software Engineering*. in *12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. 2011. 6-8 July 2011. p. 213-218.

[74]: [Scopus_64] Stroulia, E., K. Bauer, M. Craig, K. Reid, and G. Wilson. *Teaching distributed software engineering with UCOSP: the undergraduate capstone open-source project*. in *Proceedings of the 2011 community building workshop on Collaborative teaching of globally distributed software development*. 2011. ACM. p. 20-25.

[75]: [IEEE_177] Swigger, K., R. Brazile, B. Harrington, S. Peng, and F. Alpaslan. *A case study of student software teams using computer-supported software*. in *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems*. 2005. 20-20 May 2005. p. 167-173.

[76]: [IEEE_5] Swigger, K., R. Brazile, B. Harrington, P. Xiaobo, and F. Alpaslan. *Teaching Students How to Work in Global Software Development Environments*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom*. 2006. 17-20 Nov. 2006. p. 1-7.

[77]: [IEEE_153] Swigger, K., R. Brazile, F.C. Serce, G. Dafoulas, F.N. Alpaslan, and V. Lopez. *The Challenges of Teaching Students How to Work in Global Software Teams*. in *IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*. 2010. 6-9 April 2010. p. 1-30.

[78]: [ACM_22] Swigger, K., M. Hoyt, F.C. Serçe, V. Lopez, and F.N. Alpaslan, *The temporal communication behaviors of global software development student teams.* Computers in Human Behavior, 2012. **28**(2): p. 384-392.

[79]: [IEEE_142] Swigger, K., F.C. Serce, F.N. Alpaslan, R. Brazile, G. Dafoulas, and V. Lopez. *A Comparison of Team Performance Measures for Global Software Development Student Teams*. in *Fourth IEEE International Conference on Global Software Engineering, ICGSE*. 2009. 13-16 July 2009. p. 267-274.

[80]: [IEEE_193] Swigger, K., F.C. Serce, G. Dafoulas, F.N. Alpaslan, and V. Lopez. *When do distributed student teams work?* in *International Conference on Information Technology Based Higher Education and Training (ITHET)*. 2012. 21-23 June 2012. p. 1-8.

[81]: [IEEE_245] Van Solingen, R., K. Dullemond, and B. van Gameren. *Evaluating the Effectiveness of Board Game Usage to Teach GSE Dynamics*. in *6th IEEE International Conference on Global Software Engineering (ICGSE)*. 2011. 15-18 Aug. 2011. p. 166-175.

[82]: [ACM_15] Woit, D. and K. Bell. *Student communication challenges in distributed software engineering environments*. in *ACM SIGCSE Bulletin*. 2005. ACM. p. 286-290.