Global Labs - a literature review and recommendations for best practices in the domain of software systems

J Paul Gibson†

†Telecom Sud Paris, Evry, France, paul.gibson@telecom-sudparis.eu

January 15, 2020

Abstract

A major workpackage of the EU knowledge alliance project HUBLINKED was the development of Global Lab.s (GLs) for better leveraging industry-academic relations within the domain of software and ICT systems. A GL was defined as an educational unit that would put learners into *global teams* - in order to work on solving practical problems together in a laboratory setting. The *global* nature of the teams was to emphasise differences in location, language, culture, etc... The laboratory setting was to emphasise the use of a common set of tools to solve problems: learning by doing. As the focus of HUBLINKED was software and ICT, the problems that the teams would be asked to solve would be specifically in these areas. As part of the development of these GLs, we carried out a literature review of previous work/research in GLs. This paper reports on how the review was carried out, and how it contributes to the establishment of best practice of such GLs in software/ICT education.

1 Introduction

The main contribution of this paper is a review of recommended best practices for teaching using the Global Lab (GL) approach, with emphasis on Global Software Engineering (GSE). These practices are well-grounded through links to previously published research and experience reports on the subject. This paper does not directly report on our own experience of developing and deploying/teaching such GLs - this information can be found in a separate HUBLINKED deliverable **D5.2 Global Labs Experience reports**.

The bibiliographical corpus that we present in this paper is not intended to be a complete state-of-the-art: it inlcudes references to a number of much more complete literature reviews. The corpus is intended to provide evidence that our best practice recommendations are, in general, supported by the community of academics, industrialists and researchers in this area. Where there is no general agreement on these issues we will explicitly note this.

As for all such papers, it is important to state how we constructed our corpus - this is the subject of section 2 of the paper. The corpus has been deiberately structured as a tree of domains, with different layers representing the degree of specialisation/generalisation of the work being references, and the branches connecting general topics with more specialised topics.

The structured corpus is the subject of section 3. Finally, section 4 concludes the paper with a short summary of its major contributions, insights and guidelines for how the corpus can be used, maintained and extended in the future.

2 The method for building the bibliography

2.1 Guidelines

While building the bibliographical corpus for our recommended practices, we followed some well-established guidelines for systematic literature reviews.

In "Lessons from applying the systematic literature review process within the software engineering domain" [BKB+07], the authors state that "A consequence of the growing number of empirical studies in software engineering is the need to adopt systematic approaches to assessing and aggregating research outcomes in order to provide a balanced and objective summary of research evidence for a particular topic." The systematic, objective process that we adopted is based on a bottom-up structured search of the publication space in order to identify the most influential work in the area of Global Labs (GLs) for Global Software Engineering (GSE). We start with three initial research questions:

- What are the best practices for GLs?
- What are the best practices for GSE?
- What are the best practices for using GLs for teaching GSE?

In "Guidelines for snowballing in systematic literature studies and a replication in software engineering" [Woh14], the authors introduce the snowballing technique which formed the basis of our approach:

> "*Snowballing refers to using the reference list of a paper or the citations to the paper to identify additional papers. However, snowballing could benefit from not only looking at the reference lists and citations, but to complement it with a systematic way of looking at where papers are actually referenced and where papers are cited. Using the references and the citations respectively is referred to as backward and forward snowballing.*"

We start with the most specific topics of our tree-like structure, which directly correspond to the initial research questions.

Using Google Scholar as our main publication search tool, we followed the appropraite guidelines in "The role of Google Scholar in evidence reviews and its applicability to grey literature searching" [HCCK15] . In particular, we used complex regular expressions[1] for managing the use of synonyms in the titles of published work, and some simple program scripts to automatically rank and filter results. We also limited our google search to the first 100 entries found which had been published in peer-reviewed conferences or journals, or were published books with ISBNs.

Our goal was to find the most influential papers, so we examined the most cited publications (of the 100 found) and kept the 10 most highly ranked papers in each sub-category in our tree. Each newly found highly-cited paper was added to the corpus tree as low down in the

[1] See the Google instructions at About regular expressions (regex)

structure as possible (i.e. in the category which was deemed the most specialised). Following the snowballing approach, we then ranked every publication that the new member of the corpus referenced. Each of these was classified in our tree corpus structure, and if they were highly enough cited they were then queued up for addition to the corpus. In this way, our corpus was built bottom-up. As such, the ranking of the research results in each of our corpus categories is based on them having been cited by papers that directly address our research questions, and not by simply searching for the terms in oogle scholar. For example, our highest ranked publication in the `Education` category is "Situated cognition and the culture of learning"[BCD89] with a Google scholar h-index of 20836. This is not the most cited publication for "Education" in general, but it is the most cited of all the publications that we found using our bottom-up approach starting from our research questions.

2.2 Ontology - classification hierarchy and terminology

Structure of domain

In Fig. 1 we illustrate the domain structure which we used to classify the literature review in this paper.

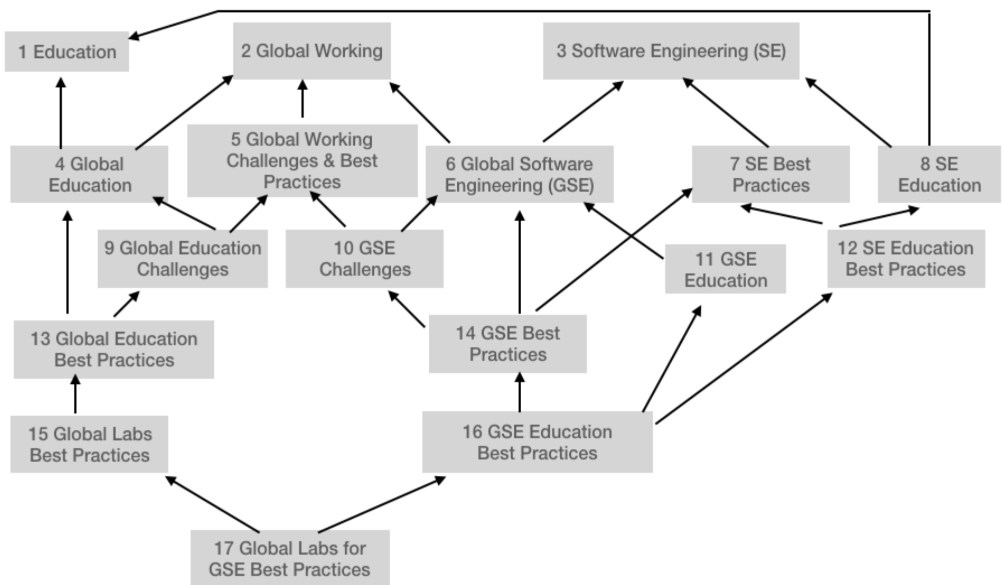

Figure 1: The literature review domain structure

The structure is tree-like acyclic graph (a forest of trees). We refer to each of the nodes of the structure to be a domain (sub)-category. The most specific catgeory is `GLs for GSE - Best Practices`. The three most generic categories are `Education`, `Global Working` and `Software engineering`. In total we have identified 17 categories into which the corpus has been structured, these have been enumerated in a top-down fashion. A directed branch from one category to another representes a specialisation relationship. For example, `Global Education` is a specialisation of `Education`. Initial categorisation of a publication in the corpus is done by title words match (using the terminology explained in the next subsection). It should be noted that some publications change category if, after detailed reading, it is clear that the content is slightly out of focus with the title words and that the article belongs in a more specific/generic category.

Terminology

In our approach we initially started our regular expression searches by using a set of synonyms. As we found new influential articles we extended our list of synonyms to include words that signalled a certain catgeory in our tree. For example, `software engineering` and `software development` were obvious synonyms; whilst `open source` was a good signal that the publication was within the software development category.

We do not claim that we have completely identified all important synonyms and signallers, but these are the key terms that we did identify in our approach:

- **Global** - on-line, online, distance, remote, distributed, correspondence, virtual, networked, WWW, web, world, dispersed, international
- **Software Engineering** - software development, software project, computer science, computing, IT, open source
- **Education** - teaching, learning, pedagogy, instruction, training
- **Working** - project, collaboration
- **Challenges** - issues, problems, risks
- **Best Practices** - recommendations, guidelines, principles, guides, tips, advice
- **Labs** - laboratories, experiments
- **Team** - teamwork, group

3 The Corpus

In this section, we present the most influential publications in each of the corpus categories (limited to 10 for each subcategory), as found using our bottom-up research question snowball approach. We start with the most general publications and finish with the most specific, following our top-down enumeration (as seen in the figure).

In each category, the publications are ranked in order of the number of citations. In some cases, we have added specialist categories not in our original taxonomy (to add structure in response to there being a significant number of important publications just for that particular topic); an example of this is the addition of the specialist topic `agility` in the software engineering categories.

3.1 Education

The highest ranked publication in the general category of education is “Situated cognition and the culture of learning”[BCD89](h-index 20836). This paper is particularly intersting from the perspective of GLs because it addresses the impact of the cultural environment on the learning process.

It should not be surprising that the “Taxonomy of educational objectives: The classification of educational goals”[Blo56] (h-index 9501) is ranked so highly: Bloom’s Taxonomy has had universal impact on teaching theory and plays a major role in establishing best educatinal practice.

From our GL experience, it is clear that students anf educators are not homogenous: “Learning and teaching styles in engineering education”[FS+88] (h-index 7462) analyses the different ways in which teachers teach and learners learn. This had an impact on the design of our GLs with respect to the use of blended learning techniques.

GLs are all about learning through doing. This type of active learning was well researched in the early 90s, and “Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports”[BE91] (h-index 6419) is an excellent well-cited introduction to the area.

Communication between students, between students and edcuators, and between students and non-academics is a major challenge in GLs. Best practice for managing communication needs to be based on well-grounded theory such as given in “Three types of interaction”[Moo89] (h-index 4428)

It would have been a surprise if “A taxonomy of problem-based learning methods”[Bar86] (h-index 3258) had been missing from the education ranking. This work has had a major impact on the incredible growth of PBL techniques across all disciplines; and GLs are considered to follow a PBL (or project-based learning) approach.

By definition, GLs use collaborative team-work across a distance. In “Computer-supported collaborative learning”[SKS06] (h-index 2989) we see the importance of tool support for such collaborative work.

A recent educational trend in team project work has been design thinking. Much of this is rooted in work from the start of the 21st century. The most widely cited reference that we found using our corpus-building approach was “Engineering design thinking, teaching, and learning”[DAE+05] (h-index 2533).

As mentioned earlier, blended learning (BL) arose out of the observation that students learn differently. This is more thorougly examined in “Blended learning systems”[Gra06] (h-index 2510), where different ways in which to build and apply BL education programmes are reviewed.

The final publication in the general education category is concerned with autonomous learning - “Toward a theory of independent learning and teaching.”[Moo73](h-index 961) - this is a central part of our GL teaching model.

Education Top 10 Ranking

[1] John Seely Brown, Allan Collins, and Paul Duguid. Situated cognition and the culture of learning. *Educational researcher*, 18(1):32–42, 1989. (h-index 20836).

[2] Benjamin Samuel Bloom. Taxonomy of educational objectives: The classification of educational goals. *Cognitive domain*, 1956. (h-index 9501).

[3] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988. (h-index 7462).

[4] Charles C Bonwell and James A Eison. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991. (h-index 6419).

[5] Michael G Moore. Three types of interaction, 1989. (h-index 4428).

[6] Howard S Barrows. A taxonomy of problem-based learning methods. *Medical education*, 20(6):481–486, 1986. (h-index 3254).

[7] Gerry Stahl, Timothy D Koschmann, and Daniel D Suthers. *Computer-supported collaborative learning.* lulu.com, 2006. (h-index 2989).

[8] Clive L Dym, Alice M Agogino, Ozgur Eris, Daniel D Frey, and Larry J Leifer. Engineering design thinking, teaching, and learning. *Journal of engineering education*, 94(1):103–120, 2005. (h-index 2533).

[9] Charles R Graham. Blended learning systems. *The handbook of blended learning*, pages 3–21, 2006. (h-index 2510).

[10] Michael Grahame Moore. Toward a theory of independent learning and teaching. *The Journal of Higher Education*, 44(9):661–679, 1973. (h-index 961).

3.2 Global Working

The most highly cited paper on global working is based on the simple observation that "Distance Matters" [Ols00] (h-index 2347) when working in teams. Olson shows that the notion of distance is concretised in many forms which can have an impact on work performance and individual behaviour.

A major problem in GLs is the sharing of distributed knowledge. In "The mutual knowledge problem and its consequences for dispersed collaboration" [Cra01] (h-index 2170), the authors examine the way in which knowledge sharing is a key part of teamwork.

GLs involve teamwork, and teams require manageent. The paper "Managing virtual teams: A review of current empirical research" [HGK05] (h-index 1242) is a highly influential publication which examines the issues of managing distributed teams.

It is important to note that there are different types of distance/separation in GLs: "The spatial, temporal, and configurational characteristics of geographic dispersion in teams" [OC07] (h-index 663).

We have already noted that knowledge-sharing is key in team work. In "Working together apart? Building a knowledge-sharing culture for global virtual teams" [ZAW04] (h-index 507), we see different techniques for constructing such knoweldge in a coherent fashion.

Recently there have been many position papers regarding Global Working. A highly cited paper is "Virtual teams research: 10 years, 10 themes, and 10 opportunities" [GMJY+15] (h-index 498), which alos provides a good review of the literature.

Global working introduces additional complexity for human resource management - "The changing nature of work and organizations: Implications for human resource management" [BN06] (h-index 464). This paper highlights the extra care that is needed in nurturing and protecting workers that are distributed globally.

In GLs, teams often divide into subteams in order to work on distinct tasks. How this division is achieved can have a major impact on group dynamics, and the success (or otherwise) of the common project: "Subgroup dynamics in internationally distributed teams: Ethnocentrism or cross-national learning?" [CH04] (h-index 340). This paper also highlights that learning between team members is a key part of working globally.

Physical separation has a major impact on communication. The paper "Why distance matters: effects on cooperation, persuasion and deception" [BM02] (h-index 305) discusses the main communication issues that may arise in a global team.

Finally, we conclude with a paper from 2007 which made some quite insigthful predictions concerning the current state of global working: "Multinational and multicultural distributed teams: A review and future agenda" [CS07] (h-index 267).

Global Working Top 10 Ranking

[1] J Olson. Distance matters. *Human-Computer Interaction*, 15, 2000. (h-index 2347).

[2] Catherine Durnell Cramton. The mutual knowledge problem and its consequences for dispersed collaboration. *Organization science*, 12(3):346–371, 2001. (h-index 2170).

[3] Guido Hertel, Susanne Geister, and Udo Konradt. Managing virtual teams: A review of current empirical research. *Human resource management review*, 15(1):69–95, 2005. (h-index 1242).

[4] Michael Boyer O'Leary and Jonathon N Cummings. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *MIS quarterly*, pages 433–452, 2007. (h-index 663).

[5] Norhayati Zakaria, Andrea Amelinckx, and David Wilemon. Working together apart? Building a knowledge-sharing culture for global virtual teams. *Creativity and innovation management*, 13(1):15–29, 2004. (h-index 507).

[6] Lucy L Gilson, M Travis Maynard, Nicole C Jones Young, Matti Vartiainen, and Marko Hakonen. Virtual teams research: 10 years, 10 themes, and 10 opportunities. *Journal of management*, 41(5):1313–1337, 2015. (h-index 498).

[7] Ronald J Burke and Eddy Ng. The changing nature of work and organizations: Implications for human resource management. *Human Resource Management Review*, 16(2):86–94, 2006. (h-index 464).

[8] Catherine Durnell Cramton and Pamela J Hinds. Subgroup dynamics in internationally distributed teams: Ethnocentrism or cross-national learning? *Research in organizational behavior*, 26:231–263, 2004. (h-index 340).

[9] Erin Bradner and Gloria Mark. Why distance matters: effects on cooperation, persuasion and deception. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 226–235. ACM, 2002. (h-index 305).

[10] Stacey L Connaughton and Marissa Shuffler. Multinational and multicultural distributed teams: A review and future agenda. *Small group research*, 38(3):387–412, 2007. (h-index 267).

3.3 Software Engineering

The publications in this category are those deemed the most influential with respect to the Global Software Engineering (GSE).

In the late 80s there was a significant move away from the traditional linear waterfall method of software development to a more iterative approach. The most impactful (spiral) model was proposed by Boehm in the paper "A spiral model of software development and enhancement"[Boe88] (h-index 6811). This model was the precursor to agile development (see later references) which is central to GSE.

As well as needing models for the software process, as above, software engineering requires a method for integrating different models in a coherent fashion. Such a unifying modeling language (UML) was the topic of "The unified software development process"[JBR99] (h-index 6428).

Agile development is now the most influential approach in modern software development. It should be no surprise that one of the most highly cited papers is the "Manifesto for agile software development"[BBVB+01] (h-index 2591).

GSE requires high levels of professionalism in order to be applied successfully. One of the most significant books on the need for professionalism and rigour is "A discipline for software engineering"[Hum95] (h-index 2027).

In the current fast-moving era of software technologies, it is a risk to forget the underlying scientific foundations of an engineering discipline. Two wonderful texts — often used as textbooks — reinforce these fundamentals: "Fundamentals of software engineering"[GJM02] (h-index 1804), and "Software engineering"[Som11] (h-index 1768).

Software engineering is largely a human endeavour. "The Mythical Man-Month: Essays on Software Engineering"[BJ95] (h-index 1025) emphasises this by reviewing the behaviour and roles of the people involved in the software development process.

In recent years, there has been a significant move towards extending agile software development in order to shorten the frequency of delivery of functional product to the clients/users. An important aspect of this type of continuous integration is the deployment of quality software: "Continuous integration: improving software quality and reducing risk"[DMG07] (h-index 778).

Open-source is one of the most successful examples of distributed software development. GLs are also concerned with encouraging innovation, and the paper "Open-source software development and distributed innovation"[KM01] (h-index 696) shows that open source can faciitate such innovative development.

Finally, we finish with a paper that addresses a major issue in software development - motivation. In "Motivation in Software Engineering: A systematic literature review"[BBH+08] (h-index 494) the authors list a number of important publications regarding how to maintain and improve the motivation of software developers. This is important with respect to working globally.

Software Engineering Top 10 Ranking

[1] Barry W Boehm. A spiral model of software development and enhancement. *Computer*, (5):61–72, 1988. (h-index 6811).

[2] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The unified software development process*, volume 1. Addison-wesley Reading, 1999. (h-index 6428).

[3] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001. (h-index 2591).

[4] Watts S Humphrey. *A discipline for software engineering*. Addison-Wesley Longman Publishing Co., Inc., 1995. (h-index 2027).

[5] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of software engineering*. Prentice Hall PTR, 2002. (h-index 1804).

[6] Ian Sommerville. *Software engineering*. Addison-Wesley/Pearson, 2011. (h-index 1768).

[7] Frederick P Brooks Jr. *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E*. Pearson Education India, 1995. (h-index 1025).

[8] Paul M Duvall, Steve Matyas, and Andrew Glover. *Continuous integration: improving software quality and reducing risk*. Pearson Education, 2007. (h-index 778).

[9] Bruce Kogut and Anca Metiu. Open-source software development and distributed innovation. *Oxford review of economic policy*, 17(2):248–264, 2001. (h-index 696).

[10] Sarah Beecham, Nathan Baddoo, Tracy Hall, Hugh Robinson, and Helen Sharp. Motivation in software engineering: A systematic literature review. *Information and software technology*, 50(9-10):860–878, 2008. (h-index 494).

3.4 Global Education (GE)

In this category, we introduce a subcategory — `Technologies` — as an area of interest and influence that merits to be considered independently. This subsection will be considered after we consider more theoretical aspects. For consistency with the rest of the corpus construction, we limit the number of cited publications in each subcategory to the 10 most highly ranked by number of citations.

GE Theory

In 2004, a highly influential paper was punlished concerned with "E-moderating: The key to teaching and learning online"[Sal04] (h-index 5296). The author provides great insight into the role of moderation in global education. Although the paper focuses on online learning,the lessons learned are generally applicable to all GE.

Five years before Salmon's paper, we see the imprortance of community in GL — "Building learning communities in cyberspace"[PP99] (h-index 4347)

Our GL approach is highly motivated by the theory of constructivism; as such, the article "Constructivism and computer-mediated communication in distance education"[JDC+95] (h-index 1784) provides strong evidence that this learning model is consistent with distance education.

Global is now — due to technological advances — synonomous with online. The paper "Towards a theory of online learning"[And04] (h-index 1155) provides a sound theoretical foundation upon which online (global) education theory can be built

There are specific issues to adults learning online; these are addressed by "Toward constructivism for adult learners in online learning environments"[Hua02] (h-index 949).

Global education changes the relationship between learners and teachers. Students must become more autonomous whilst educators must become more facilitating — "Learning in a networked world: New roles and responsibilties"[AG98] (h-index 500).

One of the earliest papers to address the changes to distance learning brought upon by technological advances was "Concepts: Toward a reconceptualization of distance education"[Sha88] (h-index 209). Although the paper was motivated by technological advances, the conceptualisation that it puts forward is theoretical in nature. A similar analysis was published in "On-line education: A study of emerging pedagogy"[Sch98] (h-index 213).

Before online education, the majority of distance learning was done through the post. Although the technologies have changed, there are many lessons that can be learned from correspondence instruction — "Theoretical aspects of correspondence instruction"[Pet71] (h-index 98).

We conclude with a paper that examines "The role of the faculty member in distance education."[Str87] (h-index 73). This paper was cited by many of the papers that were specific to gobal education for software education. For this reason it can be found in our top-10 ranking.

GE Technologies

The most cited publication on GE technologes that we found using our corpus construction approach was "Technology, e-learning and distance education"[Bat05] (h-index 3347), which provides a wide overview of a range of relevant technologies.

One of the first publications to examine the impact of the web on teaching and learning was "Growing up digital: How the web changes work, education, and the ways people learn"[Bro00] (h-index 1079). Although much of the web has changed since 2000, the insights and observations are still highly relevant today.

Even earlier to Brown's work, in 1996, Dede was one of the first researchers to predict a rapid growth in distance education supported by new technologies - "The evolution of distance education: Emerging technologies and distributed learning"[Ded96] (h-index 844).

Massive Open On-line Courses (MOOCs) have changed education forever. There are alternative views as to whether they are having a positive impact on teaching and learning. "MOOCs and open education: Implications for higher education"[YP13] (h-index 788) provides a good overview of the issues (with mostly a positive view). Whereas, "Will MOOCs destroy academia?"[Var12](h-index 284) gives a more negative opinion of the growth of such courses. Our GLs specifically avoid the MOOC model. Our current approach is to focus on active learning, problem solving and team-work in small classes. Our intention is to scale-up the GLs once we have them fine-tuned for a smaller class of particpants.

It is initially surprising that one of the most cited papers on global learning is concerned with a very simple technoloy, namely email: "Computer mediated communication for instruction: Using e-mail as a seminar."[RdH89] (h-index 133). However, on reading the paper, it is clear that email is the obvious stepping stone between correspondence learning (by traditional mail) and the use of online services. Although it is hard to imagine using email as the main delivery mechanism for learning material in the modern world, email remains one of the main tools of communication between students and teachers in global learning and, as such, this paper is a valuable addition to the corpus.

More than a quarter of a century ago the problem of time had already been identified as a major concern for global learning: "Real education in non-real time: The use of electronic message systems for instruction" [QMLB83] (h-index 131). Although this paper focuses on issues concened with assynchronous communication in collaborative working as a result of using email, the lessons it puts forward continue to be relevant for distributed collaborative work today.

Apprenticeships are a very successful pedagogic approach based on learning-by-doing and learning-from-observing. Novel communication technologies in the 1980s led to the definition of a new type of *teleapprentiship*: "Education on the electronic frontier: Teleapprentices in globally distributed educational contexts" [LRMC87] (h-index 112). The paper argues that even with limited communication infrastructure the appreniceship approach can be successfully adapted to a global learning environment. Another significant paper examining teleapprenticeships was "Teaching Teleapprenticeships: A new organizational framework for improving teacher education using electronic networks" [LWBC94] (h-index 52)

After e-mail, the world wide web (WWW) started to have significant impact on education in the late 1990s: "An experiment with WWW interactive learning in university education." [MW98] (h-index 104) The web was used mainly as a repository for sharing information (mostly reading) but also for providing self-assessment mechanisms, and interactive (albeit limited) tutorials. As we shall see in later sections on best practices, the need for more complex interaction drove the development of many more online pedagogic tools.

Global Education Top 10 Ranking

[1] Gilly Salmon. *E-moderating: The key to teaching and learning online*. Psychology Press, 2004. (h-index 5296).

[2] Rena M Palloff and Keith Pratt. *Building learning communities in cyberspace*, volume 12. San Francisco: Jossey-Bass, 1999. (h-index 4347).

[3] AW Tony Bates. *Technology, e-learning and distance education*. Routledge, 2005. (h-index 3447).

[4] David Jonassen, Mark Davidson, Mauri Collins, John Campbell, and Brenda Bannan Haag. Constructivism and computer-mediated communication in distance education. *American journal of distance education*, 9(2):7–26, 1995. (h-index 1784).

[5] Terry Anderson. Towards a theory of online learning. In Terry Anderson, editor, *Theory and practice of online learning*, volume 2, pages 109–119. Athabasca University Press, 2004. (h-index 1115).

[6] John Seely Brown. Growing up digital: How the web changes work, education, and the ways people learn. *Change: The Magazine of Higher Learning*, 32(2):11–20, 2000. (h-index 1079).

[7] Hsiu-Mei Huang. Toward constructivism for adult learners in online learning environments. *British journal of educational technology*, 33(1):27–37, 2002. (h-index 949.

[8] Chris Dede. The evolution of distance education: Emerging technologies and distributed learning. *American Journal of Distance Education*, 10(2):4–36, 1996. (h-index 844.

[9] Li Yuan and SJ Powell. MOOCs and open education: Implications for higher education. *JISC*, 2013. (h-index 788).

[10] Terry Anderson and D Randy Garrison. Learning in a networked world: New roles and responsibilties. In *Distance Learners in Higher Education: Institutional responses for quality outcomes. Madison, Wi.: Atwood.* Athabasca University Press, 1998. (h-index 500).

3.5 Global Working Challenges and Best Practices

Conflict between team members is a challenge that can be exacerbated by working globally. The paper "Out of sight, out of sync: Understanding conflict in distributed teams"[HB03] (h-index 1059) identifies two major issues:

> " *...geographically distributed teams may experience conflict as a result of two factors: The distance that separates team members and their reliance on technology to communicate and work with one another.*"

Experience shows that global teams are not as effective as local teams. This issue, together with techniques for addressing it, are considered in the book "Virtual teams that work: Creating conditions for virtual team effectiveness"[GC03] (h-index 904).

A paper from the beginning of the 21st century identifies "Five challenges to virtual team success: Lessons from Sabre, Inc."[KRG+02] (h-index 713). These challenges are classified as follows:

- Building Trust Within Virtual Teams
- Maximizing Process Gains and Minimizing Process Losses on Virtual Teams
- Overcoming Feelings of Isolation and Detachment Associated With Virtual Teamwork
- Balancing Technical and Interpersonal Skills Among Virtual Team Members
- Assessment and Recognition of Virtual Team Performance

One of the first empirical studies that analysed the effectiveness of certain practices in addressing the main challenges was published in "An empirical study of best practices in virtual teams"[LR01] (h-index 701). The study was survey-driven; and, despite there being only a small number (12) of virtual teams that participated, the results are insightful.

A major problem in team work is trusting team mates. With virtual teams the global nature of the team makes buiilding trust a significant challenge — "Building trust and collaboration in a virtual team"[Hol01] (h-index 390). The paper inculdes the conclusion:

> " *... face-to-face interaction will continue to play a very important role in our work relationships regardless of how virtual our environment may become.* "

Global companies (and projects) face addditional challenges with respect to managing personnel — "Managing the global workforce: Challenges and strategies"[RKO98] (h-index 260). This paper identifies and analyses 3 main challenges:

- deployment,

- knowledge and innovation dissemination, and
- talent identification and development.

A very insightful paper — “My Time or Yours? Managing Time Visions in Global Virtual Teams”[SSV04] (h-index 253) — examines the complex interactions between language, cultural and temporal differences in global teams. It gives concrete examples of “time language traps” and culturally different “time visions” that can have a negative impact on planning, scheduling and co-ordination.

A more scientific, theoretical, approach to global teamwork best practices is found in “Exploring Traditional and Virtual Team Members’ Best Practices A Social Cognitive Theory Perspective”[SW07] (h-index 123), where best practices in local, traditional, team work are examined within the global context. It is not surprising that local team best practices are more often than not transferable to global teams.

Some global teams are constructed in an ad-hoc manner, rather than being a more permanent entity. This can raise additional challenges. The paper “Virtual teams: An exploratory study of key challenges and strategies”[PD99] (h-index 80) explicitly addresses the issue of such ad-hoc, temporary, dynamic teams working in a global context.

Most of the more higher ranked publications in this section mention cultural differences, without considering them as a central issue. In contrast, multi-culturalism in global teams is explicitly the central theme of “Major challenges in multi-cultural virtual teams”[Vin03] (h-index 31).

Global Working Challenges Top 10 Ranking

[1] Pamela J Hinds and Diane E Bailey. Out of sight, out of sync: Understanding conflict in distributed teams. *Organization science*, 14(6):615–632, 2003. (h-index 1059).

[2] Cristina B Gibson and Susan G Cohen. *Virtual teams that work: Creating conditions for virtual team effectiveness*. John Wiley & Sons, 2003. h-index 904.

[3] Bradley L Kirkman, Benson Rosen, Cristina B Gibson, Paul E Tesluk, and Simon O McPherson. Five challenges to virtual team success: Lessons from sabre, inc. *Academy of Management Perspectives*, 16(3):67–79, 2002. h-index 713.

[4] Jeremy S Lurey and Mahesh S Raisinghani. An empirical study of best practices in virtual teams. *Information & Management*, 38(8):523–544, 2001. h-index 701.

[5] Judith A Holton. Building trust and collaboration in a virtual team. *Team performance management: an international journal*, 7(3/4):36–47, 2001. h-index 390.

[6] Karen Roberts, Ellen Ernst Kossek, and Cynthia Ozeki. Managing the global workforce: Challenges and strategies. *Academy of Management Perspectives*, 12(4):93–106, 1998. h-index 260.

[7] Carol Saunders, Craig Van Slyke, and Douglas R. Vogel. My Time or Yours? Managing Time Visions in Global Virtual Teams. *The Academy of Management Executive (1993-2005)*, 18(1):19–31, 2004. h-index 253.

[8] D Sandy Staples and Jane Webster. Exploring traditional and virtual team members best practices a social cognitive theory perspective. *Small group research*, 38(1):60–97, 2007. h-index 123.

[9] Guy Paré and Line Dubé. Virtual teams: An exploratory study of key challenges and strategies. *ICIS 1999 Proceedings*, page 50, 1999. h-index 80.

[10] Robert Vinaja. Major challenges in multi-cultural virtual teams. *Proceedings: Southwest Case Research Association*, 78541(956):341–346, 2003. h-index 31.

3.6 Global Software Engineering (GSE)

The category of global software engineering was further divided into 4 subcategories: literature review; workshop and conference proceedings; agile methods; and tool support. Before examining the main publications in each of these subcategories in more detail, we rank the most influential which deal with all other aspects of GSE in general.

The most highly cited paper on GSE is a guest editor's introduction to an early journal special issue on the subject from 2001 — "Global software development" [HM01] (h-index 1278). The authors provide a nice overview of the importance of the topic:

> "*... software development is increasingly a multisite, multicultural, globally distributed undertaking. Engineers, managers, and executives face numerous, formidable challenges on many levels, from the technical to the social and cultural.*"

Six years later, Herbsleb — an author of the previous paper — argues that GSE will be a critical part of the future: "Global software engineering: The future of socio-technical coordination" [Her07] (h-index 723). The paper examines the need for better technical coordination to manage the dependency amongst globally distributed tasks and competencies.

There are many publications that examine the cutural issues in GSE. One of the earliest, and most influential is "Managing cross-cultural issues in global software outsourcing" [KSW04] (h-index 678). A novel concept that is presented in the paper is that of "culturally-neutral software".

In the book "Global software development: managing virtual teams and environments" [Kar99] (h-index 322), emphasis is placed on the differences between traditional and virtual management. It also examines the impact of globality on the different stages of the software life cycle.

The paper "A comparison of face-to-face and virtual software development teams" [And02] (h-index 306) looked at new communication technologies that have enabled the growth of virtual global teams. They investigate

> "*... the hypotheses that both "social presence" and "media richness" associated with a communication medium used to support geographically-dispersed software development teams, will have a significant impact on team productivity, perceived interaction quality, and group process satisfaction. Results supported the predicted superiority of the face-to-face setting over the videoconferencing setting with regard to team productivity*"

An empirical study of the "Relationships among geographic dispersion, team processes, and effectiveness in software development work teams" [CW05] (h-index 194) provides results that show that "...geographic dispersion significantly and negatively relates to work processes and team effectiveness."

"Investigating cultural differences in virtual software teams" [DM02] (h-index 151) is a significant paper as it:

> "... *provides evidence for the need of computational support for effectively resolving conflicts, forming teams, dynamically allocating roles, and managing software engineering projects in culturally diverse environments.*"

Two other important publications that examine culture in GSE are: "Imparting the importance of culture to global software development" [Cas10] (h-index 40) and "Exploring the notion of cultural fit in global virtual collaborations" [Cle10] (h-index 14).

We conclude with a paper that has significantly fewer citations than the previous higher-ranked publications - "Motivation and autonomy in global software development: an empirical study" [NRB17] (h-index 9). It is included as it is a more recent publication, and it is one of the first to treat the key issues of motivation and autonomy, which are frequently mentioned as major challenges in GSE (see later sections).

GSE Literature Review

The following are the top 10 ranked citations, following our corpus building approach, that provide literature reviews on aspects of GSE:

1. "Empirical evidence in global software engineering: a systematic review" [ŠWGF10] (h-index 361).
2. "Using scrum in global software development: a systematic literature review" [HBP09] (h-index 335).
3. "Challenges and solutions in distributed software development project management: A systematic literature review" [dSCFP10] (h-index 134).
4. "Risks and risk mitigation in global software development: A tertiary study" [VBK+14] (h-index 129).
5. "The impact of global dispersion on coordination, team performance and software quality– A systematic literature review" [NDCC15] (h-index 101).
6. "Knowledge transfer challenges and mitigation strategies in global software developmentA systematic literature review and industrial validation" [NYAT13] (h-index 86).
7. "Process models in the practice of distributed software development: A systematic review of the literature" [PA10] (h-index 84).
8. "Effort estimation in global software development: A systematic literature review" [BFMU14] (h-index 38).
9. "Global software development using agile methodologies: A review of literature" [SM12] (h-index 35).
10. "Systematic literature review on agile practices in global software development" [VdSEPG18] (h-index 26).

GSE Workshops and Conferences

In this section we review the most influential workshops and conferences. Rather than ranking them by number of citations, we list them in chronological order of first appearance:

1998 - 2001 Workshop on "Software Engineering over the Internet"[Mau99, Mau00, Mau01]

2002 - 2004 Workshop on "Global software development"[Dam02, DLO03, DL04]. This workshop is a continuation of the previous workshop "Software Engineering over the Internet", the name change served to be less technology focused and to better match the more general use of the terminology amongst the wider community.

2006 - 2019 International Conference on "Global software engineering". The first conference was held in Brazil in 2006[DE06]. The most recent conference was held in Canada in 2019[CTD19, Cal19].

2007 - 2010 International Conference on "Software Engineering Approaches for Offshore and Outsourced Development"[MJ07, BJMN09, GJM09, NJMT10]

GSE Agility

Agility was one of the most common topics in research in the area of GSE, and so it merits a separate section in our corpus.

In "Can distributed software development be agile?"[RCMX06] (h-index 370), the authors

> "*... conclude that careful incorporation of agility in distributed software development environments is essential in addressing several challenges to communication, control, and trust across distributed teams.*"

A detailed study in 2009 — "Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context"[SS09](h-index 264) — examines the difficuties in being agile in a global team:

> "*Agility is increasingly being seen as an essential element underlying the effectiveness of globally distributed information systems development (ISD) teams today. However, for a variety of reasons, such teams are often unable to develop and enact agility in dealing with changing situations.*"

An excellent review of agile practices was published in "Agile practices in global software engineering — A systematic map"[JW10] (h-index 164), where the paper systematically reviews the current research literature on the use of agile practices in global software engineering (GSE) in order to highlight the main circumstances which lead to successful adoption of global agile methods.

"Could global software development benefit from agile methods?"[PL06] (h-index 144) was one of the first papers to explicitly identify the potential incompatibilities between GSE and agile approaches. In particular they examine the issue of face-to-face communication which is key to agile but apparently difficult in global teams. The solution, of course, is tool support for distributed and asynchronous communication.

An excellent emperical study comparing agile and traditional software development in a global context is published in "Agile vs. structured distributed software development: A case study" [ENF^{+}14] (h-index 101). The paper concludes by stating that:

> "*The results show no significant difference between the outcome of projects following agile processes and structured processes, suggesting that agile and structured processes can be equally effective for globally distributed development.*"

"Agile distributed software development: enacting control through media and context" [PMA12] (h-index 83) is an excellent paper whose main case study examines the different roles of formal and informal controls:

> "*demonstrates that, if appropriately applied, communication technologies can significantly support distributed, agile practices by allowing concurrent enactment of both formal and informal controls.*"

"Applying agile principles for distributed software development" [PDJ09] (h-index 67) makes a strong argument that agile methods help overcome many of the main challenges associated with GSE:

> "*Adapting these practices in a distributed environment can help distributed development tackle the challenges of cultural incompatibility, leadership struggle and lack of trust.*"

In the paper "Signs of agile trends in global software engineering research: A tertiary study" [HŠM11] (h-index 66), the authors call for more research into Agile GSE, and conclude that:

> "*... there are indications that both globalization and 'agilization' of software companies are stable trends for the future but that there is a strong need for further studies on the particular challenges that distribution of work imposes on the principles of agile development.*"

In the book "Agile software development with distributed teams: Staying agile in a global world" [Eck13] (h-index 47), Jutta Eckstein provides a very pragmatic, practical view of how to successfully integrate agile with Global in software engineering projects.

Finally, a more recent paper from 2016 — "An exploratory study in communication in Agile Global Software Development" [YGDG16] (h-index 30) —

> "*reports the results of exploratory research conducted on the impact of communication infrastructure on AGSD ... The study analyzed the perceptions of the team on the infrastructure used to support communication.*"

GSE Teamwork

After agile, teamwork was the next most discussed topic in the publications on GSE that we examined. The following are the most cited papers, ranked and categorised by the main issue that they are addressing, that we found following our research method:

1. **Visualisation tool** - "Unifying artifacts and activities in a visual tool for distributed software development teams" [FD04] (h-index 241).

2. **Time** - "The impact of time separation on coordination in global software teams: a conceptual foundation"[EC03] (h-index 195).

3. **Team configuration** - "Configuring global software teams: a multi-company analysis of project productivity, quality, and profits"[RCBH11] (h-index 85).

4. **Transition a team from local to global** - "Transitioning from a co-located to a globally-distributed software development team: A case study at Analog Devices Inc"[BF04] (h-index 56).

5. **Training teams** - "Coaching a global agile virtual team"[MCDE15] (h-index 26).

6. **Empathy** - "Group awareness in global software engineering"[LCE13] (h-index 26).

7. **Modelling Collaboration** - "Global teams: Futuristic models of collaborative work for today's software development industry" [DSB+09] (h-index 24).

8. **Diversity and conflict** - "Diversity in team composition, relationship conflict and team leader support on globally distributed virtual software development team performance" [WN15] (h-index 18).

9. **Team assessment** - "Assessing the strength of global teaming practices: A pilot study" [BRN15] (h-index 9).

10. **Social awareness and trust** - "Can social awareness foster trust building in global software teams?" [CLS13] (h-index 8).

GSE Tool support

The final theme that merited its own subcatrgory in our corpus for GSE was tool support. The following are the most cited papers that we found following our research method:

1. **Augur Tool** - In "Collaboration tools for global software engineering"[LEPV10] (h-index 211), the authors describe their Augur tool, which:

   > "*creates visual representations of both software artifacts and software development activities, and, crucially, allows developers to explore the relationship between them. Augur is designed not for managers, but for the developers participating in the software development process.*"

2. **Tool Features** - "Tools used in Global Software Engineering: A systematic mapping review"[PRVPB12](h-index 127) provides a systematic review of tools available with respect to the features they offer and concludes that:

   > "*The most common features in the GSE tools included in this study are: team activity and social awareness, support for informal communication, support for distributed knowledge management, and Interoperability with other tools.*"

3. **Distributed Communication** - "Communication tools for distributed software development teams"[TPBA07](h-index 80) is a case study driven approach which provides some general guidelines for communication tools based on observation of 3 different case-study GSE projects:

"*1. Allow teams to choose their own communication tools from a variety of options; 2. Insist on frequent communication among all members, including some synchronous interaction (telephone, chat, webconferencing); 3. Provide shared file storage to facilitate team interaction.*"

4. **Camel Tool** - In "Camel: A tool for collaborative distributed software design" [CSC+09] (h-index 41), the authors present a tool for supporting virtual software design meetings, where they address four fundamental challenges:

   "*(1) information sharing, conflict resolution and development of consensus among geographically distributed designers, (2) availability of sufficient and organizable drawing surfaces for graphical representations, (3) developing shared understanding and managing focus during the discussion and (4) appropriate capturing and storing of all design-relevant information.*"

5. **Process Support** - "Tools to support global software development processes: a survey" [PRVEP10] (h-index 41) analyses a number of tools with respect to the ISO/IEC 12207 standard by matching each tool with each GSE process that it supports:

   "*All the tools attempt to reduce the problems caused by geographic distance ... [nearly all] reduce control and coordination breakdown, ... [the majority] reduce the loss of communication... [some] support synchronous collaborative activities.... [a few] support synchronous collaborative activities ... [overall] there is a lack of connection among tools.*"

6. **Tool Challenges** - "Empirical investigation of the challenges of the existing tools used in global software development projects" [NMAH15] (h-index 32) compares results from a systematic literature review (SLR) and a questionnaire-based empirical study with respect to the main challenges in GSE tools. The main results are:

   "*The top-ranked challenges in the SLR are the inappropriate use of synchronous and asynchronous communication tools and difficulties in adopting and learning to use the existing tools. The top-ranked challenges in the questionnaire-based empirical study are the lack of awareness of existing tools used in GSD projects and the lack of support for collaboration and group decision making.*"

7. **Applying Media Synchronicity Theory** - In "Reflecting the choice and usage of communication tools in global software development projects with media synchronicity theory" [NPLP12] (h-index 32), the authors examine the impact of the application of MST in GSE. Their results indicate that:

   "*if tool use and media choice was aligned with the suggestions by MST, communication was usually found to be both efficient and effective, whereas in cases where the tool usage was suboptimal according to MST, the communicative parties did notice at least some inconvenience: they were either frustrated in the delays and inaccuracies in the replies they received, or in the inefficiency of communication.*"

8. **Software as a service (SAS)** - "Global sourcing of software development - a review of tools and services"[Mar09] (h-index 27) is one of the first papers to propose the SAS paradigm as a promising approach to optimising GSE:

   > "*SaaS and thereby the concept of a shared development environment promises to solve major issues in globally distributed software development. Shared best practises among the users of the service could further optimise globally distributed software development.*"

9. **Verification and validation** - In "Integrating early V&V support to a GSE tool integration platform"[PTE+11] (h-index 11), the authors discuss how a tool integration framework was extended to support early V&V activities via continuous integrations. They provide an interesting observation that:

   > "*When the development is done in global setting, the need for performing early V&V activities is further increased.*"

10. **Tool integration** - "A hub-and-spoke model for tool integration in distributed development"[CL16] (h-index 6) presents a light-weight tool integration model that:

    > "*can help developers fight the channel-overload and information-fragmentation problems while also increasing at the same time the awareness of the elements in their working environment.*"

All of these papers note that successful GSE is dependant on good tool support and integration for a wide range of functions.

Global Software Engineering Top 10 Ranking

[1] James D Herbsleb and Deependra Moitra. Global software development. *IEEE software*, 18(2):16–20, 2001. (h-index 1278).

[2] Pamela J Hinds and Diane E Bailey. Out of sight, out of sync: Understanding conflict in distributed teams. *Organization science*, 14(6):615–632, 2003. (h-index 1059).

[3] James D Herbsleb. Global software engineering: The future of socio-technical coordination. In *Future of Software Engineering (FOSE'07)*, pages 188–198. IEEE, 2007. (h-index 723).

[4] Srinivas Krishna, Sundeep Sahay, and Geoff Walsham. Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4):62–66, 2004. (h-index 678).

[5] Darja Šmite, Claes Wohlin, Tony Gorschek, and Robert Feldt. Empirical evidence in global software engineering: a systematic review. *Empirical software engineering*, 15(1):91–118, 2010. (h-index 361).

[6] Emam Hossain, Muhammad Ali Babar, and Hye-young Paik. Using scrum in global software development: a systematic literature review. In *2009 Fourth IEEE International Conference on Global Software Engineering*, pages 175–184. IEEE, 2009. (h-index 335).

[7] Dale Walter Karolak. *Global software development: managing virtual teams and environments*. IEEE Computer society press, 1999. (h-index 322).

[8] Hayward P Andres. A comparison of face-to-face and virtual software development teams. *Team Performance Management: An International Journal*, 8(1/2):39–48, 2002. (h-index 306).

[9] Saonee Sarker and Suprateek Sarker. Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research*, 20(3):440–461, 2009. (h-index 264).

[10] Jon Froehlich and Paul Dourish. Unifying artifacts and activities in a visual tool for distributed software development teams. In *Proceedings. 26th International Conference on Software Engineering*, pages 387–396. IEEE, 2004. (h-index 241).

3.7 SE Best Practices

It is clear that the software engineering best practices must be a fundamental part of the best practices in both software engineering, and the teaching of global software engineering. We note that the 10 papers cited here are those which have had major impact on the research into best practice of (the teaching of) global software engineering.

It is not surprising that the book "Requirements engineering: a good practice guide"[SS97] (h-index 2149) has had enormous impact on (global) software engineering research and practice. Since its publication more than 20 years ago, the guidelines for requirements engineering that it proposes have been successfully employed in a wide-variety of software and system engineering projects around the world.

"Software engineering: principles and practice"[VVVV08] (h-index 1071) is a more general book than the previous one. It highlights the importance of tooling and automation, but concludes with a pragmatic warning with reagrds their use:

> "*If the tools do not fit the procedures used within your organization, they are likely to have a far from optimal effect. Also, tools cannot make up for an ineffective development method or badly-qualified personnel. Good people deliver good products and mediocre people deliver mediocre products, irrespective of the tools they use.*"

More software is being maintained than being developed. Global software engineering best practices must inlcude software maitenance best practices. The book "Practical software maintenance: best practices for managing your software investment"[Pig96] (h-index 714) explores a basic set of "do's and don'ts", providing guidelines as to how to transition smoothly from development to maintenance.

"Scaling software agility: best practices for large enterprises"[Lef07] (h-index 450) provides a practical set of guidelines for scaling agile practices to large projects, examining diverse issues such as scaling requirements modeliing, architecture, release plans and team organisation. As agile becomes the predominant approach to all software development, including GSE, the best practices in this book are becoming increasingly relevant on a global scale.

The book "201 principles of software development"[Dav95] (h-index 297) provides a general list of principles as well as listing principles with respect to the main phases of the software life-cycle: requirements engineering, design, coding, testing, management, product assurance, and evolution. It provides the foundations for turning software development into a true engineering discipline.

"Software engineering best practices"[Jon09] (h-index 216) is based on "Lessons from Successful Projects in the Top Companies"[2]. The book focuses on software quality as the main issue:

> "*Software quality does have value, and the value increases as application sizes get bigger. In fact, without excellence in quality control, even completing a large software application is highly unlikely. Completing it on time and within budget in the absence of excellent quality control is essentially impossible.*"

Testing is an integral part to software engineering, and has an important role in establishing and maintaining trust in a global context. "Effective Software Testing: 50 Ways to Improve Your Software Testing"[Dus02] (h-index 215) contains an abundance of material and is structured in such a logical way that it is an excellent reference to all types of testing. The advice presented is applicable to all scales and types of software development and is highly relevant to testing of GSE projects.

> "*... the testing program functions as the final quality 'gate' for an application, allowing or denying the move from the comfort of the software engineering environment into the real world.*"

The most highly cited paper (rather than book) in this section is "Agile software development: adaptive systems principles and best practices"[MJ06] (h-index 203). It places emphasis on best practices for the development of complex adaptive systems (CAS). Its main contribution is the mapping of agile practices onto a 3-D project space (People, Process, and Product). The main principles it establishes are:

- Frequent Releases and Continuous Integration
- Need for Frequent Feedback
- Proactive Handling of Changes to the Project Requirements
- Loosely Controlled Development Environment
- Planning Kept to a Minimum
- Enhancing Continuous Learning and Continuous Improvement
- Emphasis on Working Software Product

"Agile Sofware Development - Best Practices for Large Software Development Projects"[SH10] (h-index 122) is another book concerned with the scalabity of agile methods. It identifies a common conflict of interest between developers and managers:

> "*This makes it so difficult to establish agile software development within an organization as a principle, despite its remarkable successes*"

[2]The subtitle of the book

The book contains guidelines for how to address such tensions when scaling up agile development processes.

This section concludes with an influential paper with European focus: "Software engineering in Europe: a study of best practices"[DLVW99] (h-index 67). Rather than just reviewing best practices, this paper examines whether best practices are actually followed by the European software industry (across many sectors and companies). It concluded that:

> "*The European software industry lags far behind the US in both awareness and application of software process improvement*"

Software Engineering Best Practices Top 10 Ranking

[1] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997. (h-index 2149).

[2] Hans Van Vliet and JC Van Vliet. *Software engineering: principles and practice*, volume 13. Citeseer, 2008. (h-index 1071).

[3] Thomas M Pigoski. *Practical software maintenance: best practices for managing your software investment*. Wiley Publishing, 1996. (h-index 714).

[4] Dean Leffingwell. *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007. (h-index 450).

[5] Alan M Davis. *201 principles of software development*. McGraw-Hill, Inc., 1995. (h-index 297).

[6] Capers Jones. *Software engineering best practices*. McGraw-Hill, Inc., 2009. (h-index 216).

[7] Elfriede Dustin. *Effective Software Testing: 50 Ways to Improve Your Software Testing*. Addison-Wesley Longman Publishing Co., Inc., 2002. (h-index 215).

[8] Peter Meso and Radhika Jain. Agile software development: adaptive systems principles and best practices. *Information systems management*, 23(3):19–30, 2006. (h-index 203).

[9] Thomas Stober and Uwe Hansmann. *Best Practices for Large Software Development Projects*. Springer, 2010. (h-index 122).

[10] Soumitra Dutta, Michael Lee, and Luk Van Wassenhove. Software engineering in europe: a study of best practices. *IEEE software*, 16(3):82–90, 1999. (h-index 67).

3.8 SE Education

In order to establish best practice in the teaching of GSE much research made reference to highly influential work on software engineering education, in general. The 10 most cited papers that we found following our corpus-building approach are listed in this section.

Project-oriented approaches to software education education have been widely reported and the most highly cited paper is a literature review: "A Review of Literature on Teaching Engineering Design Through Project-Oriented Capstone Courses"[DTMS97] (h-index 715). It is interesting that the learning objectives common to these courses are also those found in the teaching of GSE:

> "*the objective of nearly all such courses is to provide students with a real-life engineering desing experinece. Other objectives inlcude the development of interpersonal and communication skills, enhancement of student confidence, and improved university relationships with industry.*"

In the 1990s, the IEEE and ACM identified an educational need to establish a Software Engineering Body of Knowledge (SWEBOK) as an essential step towards software development becoming a professional engineering discipline. The paper "The guide to the software engineering body of knowledge" [BDA+99] (h-index 627) provides a summary and overview of the SWEBOK project and process. A major contribution is a mapping of the guide to the SWEBOK following a decompoisiton into five categories:

- Software configuration management
- Software construction
- Software design
- Software engineering infrastructure
- Software engineering management

In the last two decades there have been major advances and innovation in teaching techniques, in general, and these have been adopted and adapted by the software engineering educators. One of the most successful innovations has been the inverted classroom. The paper "Using the inverted classroom to teach software engineering" [GBH08] (h-index 444) concludes by establishing a strong link with working globally:

> "*The approach takes advantage of the benefits of both collaborative learning and distance learning while at the same time targeting the millenial student.*"

In "What knowledge is important to a software professional?" [Let00] (h-index 334), the author provides an alternative to the SWEBOK approach by surveying software professionals:

> "*Wheras most groups are basing decisions about the software engineering curriculum on the opinions of experts in the field, we are more interested in learning what subject matter practitioners themselves actually find most important in their work.*"

The most interesting observation is that practitioners suggest that educators spend less time teaching about foundational mathematics and science, in order to give more time to people skills, software processes, HCI, real-time system design, and management.

Since higher-level education has taught computer science and software engineering, there has been heated discussion concerned with the exact relationship between the two. David Parnas clarifies the situation in his seminal paper "Software engineering programs are not computer science programs" [Par99] (h-index 261). He clearly argues that:

> "*we need SE programs that follow the traditional engineering approach to professional education.*"

The paper "Software engineering education: a roadmap" [Sha00] (h-index 257) is one of the first to predict the move away from a closed-shop model of software development. It suggests that it is up to teachers to prepare students for this type of open development:

> "*educational institutions must prepare professional software developers to construct and analyze systems that are heavily constrained by non-technical considerations and that depend on independent distributed resources.*"

All students exhibit a range of different learning styles, such as sensing, visual, active or sequential. In "The ABCs of engineering education: ABET, Bloom's taxonomy, cooperative learning, and so on"[FB04] (h-index 191) the authors examine the importance of engineering teachers adapting to the learning styles of their students. The advice that they give with respect to active learning, collaborative learning and problem-based learning is particularly appropriate and insightful for the education of software engineers.

"Teaching software engineering through game design"[CC05] (h-index 182) shows the positive impact that student motivation can have on the learning experience. The development of video games is highly motivational and students met the general software development learning objectives by working in this very specific domain. The paper focuses on the game development as the main motivational factor; however, working on a team with the final delivery of a real world application (product) is also highly motivational (irrespective of whether the product is a game or not).

The highly influential paper "Improving software practice through education: Challenges and future trends"[LDHRJ+07] (h-index 156) proposes 8 outstanding challenges in the teaching of software engineering:

- making programs attractive to students,
- focusing education appropriately,
- communicating industrial reality more effectively,
- defining curricula that are forward-looking,
- providing education for existing practitioners,
- making SE education more evidence- based,
- ensuring that SE educators have the necessary background, and
- raising the prestige and quality of SE educational research.

For each challenge, they provide action items and pose open research questions. It is interesting that many of the action items are directly addressed by GSE education.

The final paper that we list in this section is "Software engineering in the academy"[Mey01] (h-index 134). The importance of working on real-world projects in teams is emphasised:

> "*An essential technique is the long-term project, which students should develop over more than a standard quarter or semestertypically over the course of a year. It should be a group project that includes aspects of analysis, design, and implementation. And it should involve the reuse, understanding, modification, and extension of existing software. The best way to achieve this last goal is to imagine the project running over several years, with each new class taking over the result of the preceding one and developing it further.*"

The design of global software engineering courses is often inspired by this point of view.

Software Engineering Education Top 10 Ranking

[1] Alan Dutson, Robert H. Todd, Spencer P. Magleby, and Carl D. Sorensen. A review of literature on teaching engineering design through project-oriented capstone courses. *Journal of Engineering Education*, 86:17–28, 1997. (h-index 715).

[2] P. Bourque, R. Dupuis, A. Abran, J.W. Moore, and L. Tripp. The guide to the software engineering body of knowledge. *Software, IEEE*, 16(6):35–44, 1999. (h-index 627).

[3] Gerald Gannod, Janet Burge, and Michael Helmick. Using the inverted classroom to teach software engineering. In *2008 ACM/IEEE 30th International Conference on Software Engineering*, pages 777–786. IEEE, 2008. (h-index 444).

[4] Timothy C Lethbridge. What knowledge is important to a software professional? *Computer*, 33(5):44–50, 2000. (h-index 334).

[5] David Lorge Parnas. Software engineering programs are not computer science programs. *Software, IEEE*, 16(6):19–30, 1999. (h-index 261).

[6] Mary Shaw. Software engineering education: a roadmap. In *International Conference on Software Engineering — Future of Software Engineering Track*, pages 371–380, 2000. (h-index 257).

[7] Richard M Felder and Rebecca Brent. The abcs of engineering education: Abet, blooms taxonomy, cooperative learning, and so on. In *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*, volume 1, 2004. (h-index 191).

[8] Kajal Claypool and Mark Claypool. Teaching software engineering through game design. In *ACM SIGCSE Bulletin*, volume 37, pages 123–127. ACM, 2005. (h-index 182).

[9] Timothy C Lethbridge, Jorge Diaz-Herrera, J Richard Jr, J Barrie Thompson, et al. Improving software practice through education: Challenges and future trends. In *Future of Software Engineering (FOSE'07)*, pages 12–28. IEEE, 2007. (h-index 156).

[10] Bertrand Meyer. Software engineering in the academy. *Computer*, 34(5):28–35, 2001. (h-index 134).

3.9 Global Education Challenges

In this section we review the most influential publications on challenges in global education. Many of these are not specifically concerned with software engineering but are very important for global labs in general.

"Issues in distance learning" [She95] (h-index 1250) is one of the first literature reviews in the area. The paper concludes with a citation which is highly relevant 25 years later:

> "*Little happens of any magnitude without administration buy-in, and the best way to achieve that is to succeed on a small level first. Put most of your effort into finding the right people rather than the most exciting technology.... Some teachers work well on camera, behind a microphone, or running a computer conference, and others do not. Find teachers who feel comfortable and work well with the media, then give*

them all of the technical support you can afford. Their job is to teach, not splice cords together or figure out why their conferencing software is misbehaving"

As highlighted in the previous paper, buy-in from different actors in the teaching support environment is key. Such buy-in is also important from other members of the faculty — "Distance education: Facing the faculty challenge"[Bow01] (h-index 470) —

> "*For a variety of reasons, faculty resist efforts to force them into distance learning. They resist individually or as a whole, often seeking the guidance of union representatives. While individual faculty members may have individual reasons to resist participating in the latest wave of distance education, there are several reasons why faculty in general resist distance education. Faculty have specifically expressed concern for the adequacy of institutional support, the change in interpersonal relations, and quality."*

The paper suggests ways in whch this can and should be addressed:

> "*Institutional support for faculty involvement in distance education is essential and should take a variety of forms to recognize the range of motivations and needs of faculty. Clearly the literature indicates that distance education classes require more faculty time than traditional courses. Institutions should recognize this and incorporate appropriate compensations when planning distance education initiatives."*

"Mapping the boundaries of distance education: Problems in defining the field"[GS87] (h-index 406) is an excellent paper that examines the problem of trying to define a fast-moving discipline:

> "*As distance education grows and becomes more technologically com- plex, attempts to provide precise definitions are certain to create anomalies and confusion. Increased awareness, a clearer identity, and an appreciation of distance education will be furthered through the listing of essential criteria that are open to refinement."*

In "Globalisation and higher education: Challenges for the 21st century"[Sco00] (h-index 343), the author identifies globalisation as both a threat and an opportunity for higher education. The paper stresses that life-long learning will become ever-more important, and that higher educational establishments will have to open up to collaborations from other acadeic and non-academic institutions in order to share responsibility for educational cooperation. The authors correctly state that this will require a long-term transformation in the way in which higher education works and is embedded in society.

An influential paper from the American Council on Education Center for Policy Analysis at the beginning of the century — "Distributed education and its challenges: An overview"[OBH01] (h-index 271) — discusses distance, distributed and e-learning. They identify three axioms that illustrate the nexus of technology and education:

- New technology affords exciting opportunities for more effective learning.
- New technology offers the prospect of reaching more learners.
- New technology will transform higher education as we know it today

The paper notes the risk in adopting new teaching technologies, as with distance education, and that sometimes not doing something new is better than doing something wrong. The authors also emphasise that

> "*the future of distributed learning - and of higher education - will not be a one-size-fits-all approach.*"

The paper "An overview of online education: Attractiveness, benefits, challenges, concerns and recommendations" [LI08] (h-index 231) is a comprehensive literature review of on-line education, including the main challenges.

In recent years, there has been much research and development for e-learning systems that support on-line learning. One of the first such systems was the synchronous e-learning system (Interwise) for online tutorials - "Replacing face-to-face tutorials by synchronous online technologies: Challenges and pedagogical implications" [Ng07] (h-index 160). The paper makes an interesting observation concerning the need for mutiple interaction methods and blended learning:

> "*some students expressed the view that a combination of online and face-to-face sessions is important for learning. The optimum mix of these two modes of learner support and the best strategies for enhancing the social dimensions of learning in an online environment are issues worth further study.*"

The book "Globalized e-learning cultural challenges" [Edm06] (h-index 164) includes many interesting sections concerned with cultural issues:

- Section 1 gives a good review of different cross-cultural education theories
- Section 2 is concerned with whether Western educational methods are appropriate in non-Western settings
- Section 3 examines the importance and difficulty of translating educational material
- Section 4 proposes concrete approaches to adapting on-line material to different cultures

The book is a very important resource for managing cross-cultural delivery of educational material.

The book "The global education terminology debate: Exploring some of the issues" [Mar07] (h-index 84) is mainly concerned with the different types of global education and the need to be precise in the use of terminology. The book focuses on fitting global education into the spectrum from multi-cultural education to human rights education. It is not primarily concerned with global eduction in the sense of on-line, but with education with global impact and teaching about global issues. It would suggest that global education, including global software engineering education, should address serious global ethical issues.

The article "What makes learning and understanding in virtual teams so difficult?" [Häk04] (h-index 72) asks

> "*whether participants in distributed learning groups are able to successfully work on a common task and achieve a type of interaction that leads them to educationally relevant higher-level discussion and learning.*"

It concludes that

"research on distributed learning groups needs to consider a complex set of variables: cognitive, social, emotional, motivational, and contextual variables interacting with each other in a systematic manner."

Global Education Challenges Top 10 Ranking

[1] Lorraine Sherry. Issues in distance learning. *International journal of educational telecommunications*, 1(4):337–365, 1995. (h-index 1250).

[2] Beverly L Bower. Distance education: Facing the faculty challenge. *Online Journal of Distance Learning Administration*, 4(2):1–6, 2001. (h-index 470).

[3] David R Garrison and Doug Shale. Mapping the boundaries of distance education: Problems in defining the field. *American Journal of Distance Education*, 1(1):7–13, 1987. (h-index 406).

[4] Peter Scott. Globalisation and higher education: Challenges for the 21st century. *Journal of studies in International Education*, 4(1):3–10, 2000. (h-index 343).

[5] Diana Oblinger, Carole A Barone, and Brian L Hawkins. *Distributed education and its challenges: An overview*, volume 1. American Council on Education Washington, DC, 2001. (h-index 271).

[6] Chi-Sing Li and Beverly Irby. An overview of online education: Attractiveness, benefits, challenges, concerns and recommendations. *College Student Journal*, 42(2), 2008. (h-index 231).

[7] Kwok Chi Ng. Replacing face-to-face tutorials by synchronous online technologies: Challenges and pedagogical implications. *The International Review of Research in Open and Distributed Learning*, 8(1), 2007. (h-index 160).

[8] Andrea Edmundson. *Globalized e-learning cultural challenges*. IGI Global, 2006. (h-index 164).

[9] Harriet Marshall. The global education terminology debate: Exploring some of the issues. *The SAGE handbook of research in international education*, pages 38–50, 2007. (h-index 84).

[10] Päivi Häkkinen. What makes learning and understanding in virtual teams so difficult? *CyberPsychology & behavior*, 7(2):201–206, 2004. (h-index 72).

3.10 Global SE Challenges

The paper "RE challenges in multi-site software development organisations" [DZ03] (h-index 363) reports on a field study that investigated requirements engineering challenges introduced by the geographical distribution of stakeholders in a distributed organisation, and the additional challenges introduced:

"Requirements engineering is a task difficult enough when done locally - but it is even more difficult when cross-functional stakeholder groups specify requirements across cultural, language and time zone boundaries."

The paper concludes with recommended practices for managing:

- cultural diversity
- inadequate communication
- ineffective knowledge management
- time differences

In "Global software development challenges: A case study on temporal, geographical and socio-cultural distance"[HCAF06] (h-index 323), the authors present the 3 main challenges related to GSE - temporal, geographical and socio-cultural distance. The research is based on workshop discussions and qualitative interviews with members of the software industry.

"Challenges and improvements in distributed software development: A systematic review"[JPV09] (h-index 258) presents the findings of a systematic review of the literature related to the challenges concerning Distributed Software Development. It validates the previous research that identified 3 types of distance as being key issues. It extends that result by identifying other serious issues such as knowledge management. The report concludes with a warning that there is no one-size-fits-all solution to these challenges:

> "*However, every organization has concrete needs which basically depend on its distribution characteristics, its activity and the tools it employs. These factors therefore cause this subject to be extremely wide-ranging, and lead to the necessity of adapting both the technical and organizational procedures, according to each organizations specific needs.*"

Another extensive survey paper - "Global software development and collaboration: barriers and solutions"[NBR10] (h-index 238) - reviews the global software development literature, highlighting collaboration problems experienced by a cross-section of organisations in twenty-six studies. The study found:

> "*that the key barriers to collaboration are geographic, temporal, cultural, and linguistic distance; the primary solutions to overcoming these barriers include site visits, synchronous communication technology, and knowledge sharing infrastructure to capture implicit knowledge and make it explicit.*"

The paper "Challenges of global software development"[MH01](h-index 176), classifies the challenges into 3 different types:

- lack of or differences in infrastructure in different development locations
- interdependencies among work items and difficulties of coordination.
- issues related to communication across sites

In summary they suggest that:

> "*de-coupling of work and creating a virtual site as two approaches that may reduce or eliminate problems associated with distributed work.*"

The papr "Challenges of project management in global software development: A client-vendor analysis" [NMA$^+$16] (h-index 85) combines a comprehensive literature review with extensive industry feedback in order to compare the perceived challenges with the actual issues that arise in GSE projects. The paper also distinguishes between client-oriented and vendor-oriented challenges. This separation provides nice insight into the challenges which are common to both and those which are visible to only one side of the client-vendor relationship.

Although the paper "Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities" [HP05] (h-index 84) appears to be education oriented, it does an excellent job of identifying the main GSE challenges, albeit through the perspective of what this means for the teaching of software engineering. The paper identifies specific areas of competency that need to be reinforced due to GSE challenges:

> "*organization and process engineering, system and product family architectures ,integration techniques and technologies, product and productline management, and distributed project management.*"

The issue of estimating the complexity or difficulty of a GSE project is addresed in "Global software engineering: Identifying challenges is important and providing solutions is even better" [BL14] (h-index 45). The paper defines ten heuristics for carefully planning, organizing, and managing GSE projects:

- Plan the distribution deliberately
- Start locally and grow globally
- Carefully select employees and prepare them for their tasks
- Establish a common goal
- Further the exchange of employees
- Provide a suitable IT infrastructure
- Define clear communication structures and a global escalation path
- Utilize time zone differences
- Pay attention to clear requirements and domain knowledge
- Use an iterative process and foster continuous improvement

As this lists shows, many of these guidlelines are not specific to GSE, and are derived from software engineering best practice.

Cultural and linguistic distance have been identified as major issues in GSE. The paper "The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges" [AM10] (h-index 38) discusses the impact of the cultural and linguistic aspects of global software on the GSE process. The research presented is based on the standard Hofstede cultural dimensions model which describes five dimensions of differences (value perspectives) between national cultures -

- Power Distance

- Individualism vs. Collectivism
- Masculinity vs. Femininity
- Uncertainty Avoidance
- Long-Term vs. Short-Term orientation

The paper concludes that this 5-dimensional model may not be rich enough for reasoning about GSE:

> "*there is more and more evidence that the Hofstede cultural model is not enough to understand the impact of culture on software.*"

The final paper in this section is a more recent review of Agile communication challenges — "Agile global software development communication challenges: A systematic review" [AG14] h-index 32). It uses a Systematic Literature Review (SLR) to address the following question:

> "*What is currently known about the communication challenges in the context of agile GSD?*"

The paper answers the question by reviewing the challenges or factors that limit communication in GSE; and reviewing the strategies, techniques or practices that are being used to deal with these challenges to enhance communication in agile GSE.

Global Software Engineering Challenges Top 10 Ranking

[1] Daniela E Damian and Didar Zowghi. RE challenges in multi-site software development organisations. *Requirements engineering*, 8(3):149–160, 2003. (h-index 363).

[2] Helena Holmstrom, Eoin Ó Conchúir, J Agerfalk, and Brian Fitzgerald. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, pages 3–11. IEEE, 2006. (h-index 326).

[3] Miguel Jiménez, Mario Piattini, and Aurora Vizcaíno. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, 2009:3, 2009. (h-index 258).

[4] John Noll, Sarah Beecham, and Ita Richardson. Global software development and collaboration: barriers and solutions. *ACM inroads*, 1(3):66–78, 2010. (h-index 238).

[5] Audris Mockus and James Herbsleb. Challenges of global software development. In *Proceedings seventh international software metrics symposium*, pages 182–184. IEEE, 2001. (h-index 176).

[6] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, Mohammed Rehan Riaz, Kanaan Faisal, Narciso Cerpa, Siffat Ullah Khan, and Ita Richardson. Challenges of project management in global software development: A client-vendor analysis. *Information and Software Technology*, 80:1–19, 2016. (h-index 85).

[7] Matthew J Hawthorne and Dewayne E Perry. Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities. In *International Conference on Software Engineering*, pages 166–185. Springer, 2005. (h-index 84).

[8] Muhammad Ali Babar and Christian Lescher. Global software engineering: Identifying challenges is important and providing solutions is even better. *Information and Software Technology*, 56(1):1–5, 2014. (h-index 45).

[9] Sameer Abufardeh and Kenneth Magel. The impact of global software cultural and linguistic aspects on global software development process (gsd): Issues and challenges. In *4th International Conference on New Trends in Information Science and Service Science*, pages 133–138. IEEE, 2010. (h-index 38).

[10] Yehia Ibrahim Alzoubi and Asif Qumer Gill. Agile global software development communication challenges: A systematic review. In *Proceedings-Pacific Asia Conference on Information Systems, PACIS 2014*, 2014. (h-index 32).

3.11 Global SE Education

The most cited paper in this section is one of the earlest to be published on the subject of teaching GSE using teams of distributed students — "Analysis of the effectiveness of global virtual teams in software engineering projects"[ES03] (h-index 111). The paper reports on the use of globally distributed (between North America and Asia) working in virtual teams. The software engineering project was focused on requirements engineering but, as we have seen from previously cited work, this is a key part of all software engineering projects and should provide a good basis upon which to analyse the virtual team work for software engineering education in general. The study produced two very interesting observations:

- **Role of technology** -

  > "*First, as technology becomes easy to use and the structure of the project increased, the effectiveness and satisfaction level of the team members as well as the efficiency of the software engineering process increased.*"

- **Importance of trust** -

  > "*Trust between the peer teams has significant positive association with the virtual team project experience of the students as well as the learning effectiveness of the team members. An increase in trust also increases the efficiency of the software engineering process itself.*"

The paper "Teaching practical software engineering and global software engineering: evaluation and comparison"[PTT06a] (h-index 65) reports on a blended learning approach to teaching GSE:

> "*Our approach was to combine and synchronize class teaching about SW engineering methods and processes with actual SW development work in a setting designed to simulate a small SW company.*"

They concluded that

> "*providing adequate global SE education requires very significant effort on instructors, before the class (planning) and during the class. This is well worth it, evidenced by class success and very positive student feedback.*"

In "Distributed development: an education perspective on the global studio project" [RMMK06] (h-index 63) the authors report on the Global Studio Project (GSP), which has "organized the work of Software Engineering and Computer Science student teams from five Universities in four countries into a single global project". The paper emphasises the need to work on real projects with real clients. One of its most valuable observations is on the educational value of improvisation:

> "*We believe that the most significant educational value of the GSP was in the area of improvisation. When students joined the project, they had already experienced most or all of the traditional class work associated with software engineering programs. This gave most of them at least a basic a repertoire of project process models, team organizations, architectural and software design patterns and testing strategies. While stringing these together in a single, complex project provided some educational value in itself, the real discoveries came from the frustrations and anxiety associated with the need to research, plan, communicate and respond to frequent changes quickly in the GSD environment.*"

The use of agile methods in GSE projects for students is reported in "Teaching students global software engineering skills using distributed scrum" [PLD+13] (h-index 46). The paper's conclusions are very positive:

> "*Our analysis indicates that the Scrum method, along with supporting collaboration practices and tools, supports the learning of im- portant GSE competencies, such as distributed communication and teamwork, building and maintaining trust, using appropriate collaboration tools, and inter-cultural collaboration.*"

A recently published literature review — "Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review" [CBB+15] (h-index 41) — references a substantial amount of research on GSE education. Although the implementation of a GSE education module can be costly, it has been shown to have significant benefits in a large number of higher level institutions around the world:

> "*Teaching software engineering is difficult in a co-located setting, and as this review shows, teaching GSE-Ed courses comes with a considerable overhead, mainly due to distance issues. Teaching GSE-Ed in university settings is not for the faint hearted; yet, as the 82 studies in this SLR testify; many universities are doing just that.*"

There are different ways in which GSE can be taught. Two different approaches are compared in "Globalizing software development in the local classroom" [RMM+10] (h-index 41):

- students shadow the development of a real-life GSD project.
- collaborative virtual team software testing was carried out by the students

Overall, their findings

> "*confirm that mimicking real work settings has educational benefits for problem-based learning environments.*"

Two projects that report on GSE education - PASTICS and Runestone - are reviewed in "Reflections on international projects in undergraduate CS education"[DBP99] (h-index 33). The paper identifies important criteria for validating the use of global projects for teaching GSE -

- The syllabus is covered at least as well as through 'conventional' methods.
- The actual time the students spend on the course is related to the 'allotted' time.
- The time staff spend on the course is related to the size of the course and is comparable to other ways of delivering the course.
- The cost, apart from staff time, of running a class is not higher than other forms.
- The course contributes to the personal development of the students.
- The form is motivating to students.

The paper concludes by stating:

> "*The experiences on the PASTICS and Runestone projects suggest that the international project form itself is motivating to students, but that its efficacy is affected less by technical skills (which we might expect our students to have) than by social and communications skills* "

The importance of having real customers in a GSE student project is highlighted in "Customers' role in teaching distributed software development"[BCŽ+10] (h-index 27). The paper poses some interesting questions that require further research:

> "*What are the advantages/disadvantages in academia, where the same people play the role of both customer and supervisor? What are the characteristics of university customers? What are the overall characteristics of such projects? How does the customer location influence the project work (co-located with the main site; no co-location)? How does the supervisor role change, when the customer is external? Can a supervisor influence the customer? Can he have influence on students? What are the motives of industry customers? What are the motives' implications on the project?* "

The challenge of encouraging companies to be involved in GSE student projects is discussed in "Preparing tomorrow's software engineers for work in a global environment"[BCB+17] (h-index 21).

> "*GSE projects often suffer from a dearth of willing industry collaborators, compromising many attempts to emulate real-world settings. This situation has several causes. For example, companies might not want to participate because they fear that studentswho typically take multiple courses at the same time might not give the GSE project a high priority. Also, students generally dont have the skills or awareness of impor- tant issues that professionals have. Nonetheless, companies should also look at*

the advantages of participating in universities GSE projects. After all, these students could be their future employees. GSE education is a difficult endeavor that schools undertake with limited resources. Companies could help by supplying resources such as tools, expertise, or project involvement.'

A question that often arises is if agile SE is better taught using same-site or cross-site teams. An initial invetsigation of the issues, within the context of agile development, is reported in "Learning global agile software engineering using same-site and cross-site teams" [PBL+15] (h-index 21). The research approach was to partition the students into those who worked locally and those who worked globally. Then, half-way through the projects, the teams were changed so that every student worked in both types of project. Their overall conclusion was surprising:

> *"the students did not report significant differences between working in local versus global teams in regards to communication, teamwork and conformance to Scrum practices used."*

This supports the view that agile methods, such as Scrum, can mitigate the problems associated with GSE.

Global Software Engineering Education Top 10 Ranking

[1] H Keith Edwards and Varadharajan Sridhar. Analysis of the effectiveness of global virtual teams in software engineering projects. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pages 9–pp. IEEE, 2003. (h-index 111).

[2] Dragutin Petkovic, Gary Thompson, and Rainer Todtenhoefer. Teaching practical software engineering and global software engineering: evaluation and comparison. *ACM SIGCSE Bulletin*, 38(3):294–298, 2006. (h-index 65).

[3] Ita Richardson, Allen E Milewski, Neel Mullick, and Patrick Keil. Distributed development: an education perspective on the global studio project. In *Proceedings of the 28th international conference on Software engineering*, pages 679–684. ACM, 2006. (h-index 63).

[4] Maria Paasivaara, Casper Lassenius, Daniela Damian, Petteri Räty, and Adrian Schröter. Teaching students global software engineering skills using distributed scrum. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1128–1137. IEEE, 2013. (h-index 46).

[5] Tony Clear, Sarah Beecham, John Barr, Mats Daniels, Roger McDermott, Michael Oudshoorn, Airina Savickaite, and John Noll. Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. In *Proceedings of the 2015 ITiCSE on Working Group Reports*, pages 1–39. ACM, 2015. (h-index 41).

[6] Ita Richardson, Sarah Moore, Alan Malone, Valentine Casey, and Dolores Zage. Globalizing software development in the local classroom. In *IT Outsourcing: Concepts, Methodologies, Tools, and Applications*, pages 1534–1556. IGI Global, 2010. (h-index 41).

[7] Mats Daniels, Anders Berglund, and Marian Petre. Reflections on international projects in undergraduate cs education. *Computer Science Education*, 9(3):256–267, 1999. (h-index 33).

[8] Ivana Bosnic, Igor Cavrak, Mario Žagar, Rikard Land, and Ivica Crnkovic. Customers' role in teaching distributed software development. In *2010 23rd IEEE Conference on Software Engineering Education and Training*, pages 73–80. IEEE, 2010. (h-index 27).

[9] Sarah Beecham, Tony Clear, John Barr, Mats Daniels, Michael Oudshoorn, and John Noll. Preparing tomorrow's software engineers for work in a global environment. *IEEE Software*, 34(1):9–12, 2017. (h-index 21).

[10] Maria Paasivaara, Kelly Blincoe, Casper Lassenius, Daniela Damian, Jyoti Sheoran, Francis Harrison, Prashant Chhabra, Aminah Yussuf, and Veikko Isotalo. Learning global agile software engineering using same-site and cross-site teams. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 285–294. IEEE, 2015. (h-index 21).

3.12 SE Education Best Practices

In order to establish best practice in the teaching of GSE, it is a good first step to identify best practice in the teaching of software engineering in general. This section reports on the previous publications in this area that have had most impact on establishing GSE education best practices.

In "Simulation in software engineering training" [DL00] (h-index 213), we see the importance of real world simulation when students cannot be given real responsibilities in the real world. In particular, it reports on a simulator for project management (SESAM) which has been used succesfully over a number of years.

Best practices are often implicit in curriculum recommendations; in "A Master of Software Engineering Curriculum: Recommendations from the Software Engineering Institute" [FG89] (h-index 69) we see such recommendations for a masters level programme in Software Engineering. This paper is over 30 years old, but it predicts the current move towards innovative SE education outside the classroom:

> "*The software community generally agrees that experience plays a leading role in the development of a good software engineer. Some persons have suggested that software engineering look to medicine for educational models such as internships, residencies, and teaching hospitals. Thus, the SEI will continue investigating the structure of the MSE programs experiential component.*"

A similar paper to the previous one focuses on undergraduate education - "SE2004: Recommendations for undergraduate software engineering curricula" [LLJS+06] (h-index 54). It proposes a body of knowledge that an undergraduate should be expected to cover, in contrast to the SWEBOK which identifies what a practitioner with years of experience should know.

"Collaborating with industry: strategies for an undergraduate software engineering program" [Rei06] (h-index 53) is motivated by the need to work in the real world in order to learn:

> "*Even the best written text books and most well designed laboratory exercises on their own cannot account for the preparation needed to become effective members of professional software development teams.*"

The paper reports on the use of commercially sponsored capstone SE team projects in order to provide students with such a learning experience.

As we have already seen, using projects is in intself considered good practice in the teaching of SE. In "Good practices for educational software engineering projects"[VDDAS07] (h-index 47), the authors address the need for good practices when following such an approach. A major issue with group projects is that of *free-riders* — students who contribute little whilst benefiting from the work of others in the team. This paper makes 3 concrete recommedations for managing this probllem:

- Time logging
- Continuous evaluation
- Working with pre-defined processes

"The relevance of software education: A survey and some recommendations"[Let98] (h-index 40) reports on an industrial survey where the authors:

> "*present some advice to companies regarding topics in which they should train employees; we compare this advice to a list of skills that employers have indicated they wish employees to possess.*"

The paper's conclusion reflects the move towards life-long learning rather than short-term technology driven curriculum:

> "*In particular, employers should think more about longer-term skills rather than particular tools or languages, unless they are hiring staff merely for short term contracts.*"

"Teaching software engineering through real-life projects to bridge school and industry"[Son96] (h-index 30) continues the theme of many of the previous papers by emphasising the need for students to work on real world projects in teams. This is one of the earliest publications to report on an educational experiment where the authors:

> "*present our experience of exposing graduate students to a real-time plant monitoring and control software developmentproject and show how the software engineering process has been customized to educate them and satisfy the user requirementsat the same time.*"

In their study the authors were interested in the potential conflict of interest between students meeting learning objectives, and meeting client requirements; however, they stress that:

> "*the software engineering process can be appropriately customized to educate students, if it is required, without failing to meet client requests. By customizing software engineering processes according to the needs of students, university software education may provide a more pragmatic real-world working environment for software engineering students and contribute to their successin industries.*"

In "Teaching software engineering: A practical approach"[Dah10] (h-index 23), the authors motivate their research by stating that "it is a major challenge to integrate applied methodology and theory into the practice of software development." The paper proposes a blended learning approach and emphasises the importance of coherent integration of theory and practice:

> "*Only when the bricks and mortar of software engineering have delivered a solid foundation, is it then that software engineering as a course will be delivered in a true and meaningful sense.*"

The innovative teaching approach presented in the paper is the use of *lecture demos*. These are neither traditional lectures, nor traditional laboratory practical sessions. Rather, they are scheduled lectures in which the teacher demonstrates the use of some tool or technique whose operation is based on, or implements, some more theoretical material that the students have already seen. Feedback from the students was very postive.

The book "Software Engineering: Effective Teaching and Learning Approaches and Practices: Effective Teaching and Learning Approaches and Practices" [Ell08] (h-index 20) provides many guidelines for a range of key practices in teaching software engineering - student assessment and learning, innovative teaching methods, and educational technology. It has contributions from many leading educators in the field.

In many capstone projects, students are required to follow a certain software development process. The paper "The software enterprise: Practicing best practices in software engineering education" [Gar08] (h-index 17) argues that more emphasis needs to be placed on the teaching of process rather than just its application. They put forward the view that software processes implicitly embody knowledge of best practice and that best practice in the teaching of a software engineering capstone module must incorporate teaching of best practices in software processes.

The paper concludes by reviewing the reason for the success of their "Enterprise" approach:

> "*Instead of setting a broad foundation and then driving deep into narrow area of specialty, the Enterprise instead looks at the emergent expertise gained by junior professionals to create a new pedagogical model for software engineering that mimics exposure and teamwork found in industry.*"

Software Engineering Education Best Practices Top 10 Ranking

[1] Anke Drappa and Jochen Ludewig. Simulation in software engineering training. In *Proceedings of the 22nd international conference on Software engineering*, pages 199–208. ACM, 2000. (h-index 213).

[2] Gary A. Ford and Norman E. Gibbs. A master of software engineering curriculum: Recommendations from the software engineering institute. *Computer*, 22:59–71, September 1989. (h-index 69).

[3] Timothy C Lethbridge, Richard J LeBlanc Jr, Ann E Kelley Sobel, Thomas B Hilburn, and Jorge L Diaz-Herrera. Se2004: Recommendations for undergraduate software engineering curricula. *IEEE software*, 23(6):19–25, 2006. (h-index 54).

[4] Thomas J Reichlmay. Collaborating with industry: strategies for an undergraduate software engineering program. In *Proceedings of the 2006 international workshop on Summit on software engineering education*, pages 13–16. ACM, 2006. (h-index 53).

[5] Louwarnoud Van Der Duim, Jesper Andersson, and Marco Sinnema. Good practices for educational software engineering projects. In *29th International Conference on Software Engineering (ICSE'07)*, pages 698–707. IEEE, 2007. (h-index 47).

[6] Timothy C Lethbridge. The relevance of software education: A survey and some recommendations. *Annals of Software Engineering*, 6(1-4):91–110, 1998. (h-index 40).

[7] Ki-Sang Song. Teaching software engineering through real-life projects to bridge school and industry. *ACM SIGCSE Bulletin*, 28(4):59–64, 1996. (h-index 30).

[8] Deepak Dahiya. Teaching software engineering: A practical approach. *ACM SIGSOFT Software Engineering Notes*, 35(2):1–5, 2010. (h-index 23).

[9] Heidi JC Ellis. *Software Engineering: Effective Teaching and Learning Approaches and Practices: Effective Teaching and Learning Approaches and Practices*. IGI Global, 2008. (h-index 20).

[10] K.A. Gary. The software enterprise: Practicing best practices in software engineering education. *International Journal of Engineering Education*, 24(4):705, 2008. (h-index 17).

3.13 Global Education Best Practices

In this section, we focus on the most influential publications in the general area of global education best practices. These publications were identfied by examining the citations in the more specific publications on best practice in global software engineering education.

The book "Building online learning communities: Effective strategies for the virtual classroom" [PP07] (h-index 1214) provides a practical, hands-on guide, using illustrative case studies, vignettes, and examples from a range of successful online courses. The authors offer proven strategies for handling GE challenges, including:

- Engaging students in the formation of an online learning community.
- Establishing a sense of presence online.
- Maximizing participation.
- Developing effective courses that include collaboration and reflection.
- Assessing student performance.

The paper "Critical success factors in online education" [VL00] (h-index 988), is one of the first publications to report on a survey conducted amongst students enrolled in one online management course. It identifies three critical success factors in online delivery: technology, the instructor and the previous use of the technology from a student's perspective. They also show that:

> "*the lecturer will continue to play a central role in online education, albeit his or her role will become one of a learning catalyst and knowledge navigator.*"

A good literature review of best pracitices in GE is found in the paper "Towards best practices in online learning and teaching in higher education" [KK10] (h-index 275). The cited publications focus on the transition problem, and the need for:

> "*effective strategies to enhance faculty success in their transition from traditional pedagogical platforms to online learning and teaching.*"

Much of the recent research on best practices for GE focus on MOOCS. A highly cited paper is "MOOC pedagogy: gleaning good practice from existing MOOCs"[Bal14] (h-index 257), and it provides some good pedagogical practices for MOOCs, whilst highlighting areas for improvement. The paper concludes with a warning concerning course completion targets

> "*the designers of xMOOCs can do better if they focus more on promoting deeper learning than on designing easy assessments that encourage course completion. The current trends may help improve completion rates in the short term, but harm the reputation and potential of MOOCs in the longer term.*"

The impact of global education on educators should not be underestimated. The paper "Effective pedagogical practices for online teaching: Perception of experienced instructors"[BC09] (h-index 194) provides a teacher-oriented persepective and recommendations:

> "*Instructors in this study described the need for flexibility and technological skills in dealing with system delays, online learning system platform accessibility, e-mail reliability, and being able to work through communicating with students using only the written medium. These are all issues that may test the patience of both faculty and students. Adapting to new technological tools, including state-of- the-art software and innovative hardware, is important to successful online teaching. Revisions and improvements to online learning systems can be expected to continue as online teaching evolves. The challenge for instructors will be to continue to integrate effective pedagogical practices as these technological tools evolve.*"

Global learning often leads to a tension between the teachers and the learners. The paper "Best practices in predicting and encouraging student persistence and achievement online"[MF08] (h-index 138) reports on how the faculty-student realtionship can be improved within the context of global education. Firstly, there is a proposal for *pair-teaching*:

> "*faculty members who are new to the online environment would benefit from being paired with an effective online instructor for an initial course. In this way, the novice faculty member can observe the technological, managerial, social, and pedagogical roles of faculty and how and when to engage those roles online.*"

Secondly, they provide useful advice conncerning adapting to students' needs through blended learning:

> "*students need active faculty involvement early in the course in order to understand the course layout, assignments, and expectations. Identifying weaker students through pre-assessment and other less assertive students may help faculty identify students who will need individual assistance early on. Students in this environment will benefit from the faculty practice of multiple instructions, multiple times, in multiple ways.*"

"Best practices: A triangulated support approach in transitioning faculty to online teaching"[CPW05] (h-index 109) is another influential publication that examines the transition from traditional teaching to on-line teaching. Much like the previous paper proposed academic support through *pair-teaching*, this paper advises on the use of teaching mentors:

"several key faculty members who were early adopters in teaching distance-education courses took the lead in providing individualized and structured support to their colleagues. These faculty members, who came to be known during professional development activities as the Professional Writing faculty mentors, were instrumental in providing support to colleagues who were beginning to move their traditional instructional practices to the online environment."

The use of open-ended group projects is presented in "Engineering education research in practice: Evolving use of open ended group projects as a pedagogical strategy for developing skills in global collaboration"[DCPC10] (h-index 93). Although the paper identifies good practices for this pedgogical approach, it also highlights the outstanding issues that require further research:

"building and sustaining common motivation across student cohorts; managing differing courses and outcomes; managing perceptions about the course; providing meaningful learning experiences to groups of students with differing competencies; and accommodating linguistic and cultural differences."

In distance education there are four main actors to consider: the instructor, the learner, the technology and the organization "Distance education: Pedagogy and best practices in the new millennium"[MK03] (h-index 86) treats each of these view points in turn, and the interdependencies between them, with focus on the need for educational platform support:

"The instructional design of the course should determine how these platforms should be used. Problem-based and case-based scenarios, in addition the use of papers, collaborative projects, portfolios, and other forms of authentic assessment increases the authenticity of the learning experience of. The efficacy of the platforms used rests in the Instructional Design of the course."

One of the earliest publications concerning best practices for on-line practical work is found in the paper "Contribution to the definition of best practices for the implementation of remote experimentation solutions"[VGS01] (h-index 14), where they recommend "the advantages of an open and multiplatform solution based on Internet standards."

Global Education Best Practices Top 10 Ranking

[1] Rena M Palloff and Keith Pratt. *Building online learning communities: Effective strategies for the virtual classroom*. John Wiley & Sons, 2007. (h-index 1214).

[2] Thierry Volery and Deborah Lord. Critical success factors in online education. *International journal of educational management*, 14(5):216–223, 2000. (h-index 988).

[3] Jared Keengwe and Terry T Kidd. Towards best practices in online learning and teaching in higher education. *MERLOT Journal of Online Learning and Teaching*, 6(2):533–541, 2010. (h-index 275).

[4] Maha Bali. Mooc pedagogy: gleaning good practice from existing moocs. *Journal of Online Learning and Teaching*, 10(1):44, 2014. (h-index 257).

[5] Craig J Bailey and Karen A Card. Effective pedagogical practices for online teaching: Perception of experienced instructors. *The Internet and Higher Education*, 12(3-4):152–155, 2009. (h-index 194).

[6] Libby V Morris and Catherine L Finnegan. Best practices in predicting and encouraging student persistence and achievement online. *Journal of College Student Retention: Research, Theory & Practice*, 10(1):55–64, 2008. (h-index 138).

[7] David Covington, Donna Petherbridge, and Sarah Egan Warren. Best practices: A triangulated support approach in transitioning faculty to online teaching. *Online Journal of Distance Learning Administration*, 8(1):3–14, 2005. (h-index 109).

[8] Mats Daniels, Åsa Cajander, Arnold Pears, and Tony Clear. Engineering education research in practice: Evolving use of open ended group projects as a pedagogical strategy for developing skills in global collaboration. *International journal of engineering education*, 26(4):795–806, 2010. (h-index 93).

[9] Thomas Miller and Frederick King. Distance education: Pedagogy and best practices in the new millennium. *Int. Leadership in Education*, 6(3):283–297, 2003. (h-index 86).

[10] Xavier Vilalta, Denis Gillet, and Christophe Salzmann. Contribution to the definition of best practices for the implementation of remote experimentation solutions. In *IFAC Workshop on Internet Based Control Education (IBCE'01)*, number CONF, 2001. (h-index 14).

3.14 Global Software Engineering Best Practices

This section focuses on the industrial perspective of how best to work globally on software engineering projects.

The most influential paper in this section addresses the issue of distance and how best to manage the different distances between team members: "Tactical approaches for alleviating distance in global software development"[CA01] (h-index 903). The paper proposes cocnrete tactics to follow in order to reduce:

- intensive collaboration
- cultural distance
- temporal distance

"Global software development at siemens: experience from nine projects"[HPB05] (h-index 303) gives a very pragmatic, industry-oriented viw of GSE based on a range of significnt case-studies. It concludes with a model based on *dimensions of coordination*:

> "*It seems clear there is no single right model of successful collaboration across sites. We find it useful to think about the possibilities in terms of dimensions of coordination. There are many ways to coordinate work, including shared processes; shared, detailed, project management; modular product structure; shared past experience; background knowledge, e.g., of the domain; and finally, planned communication regimes, including status meetings and technical reviews.*"

The contribution of the paper "Agile practices reduce distance in global software development" [HFÅC06] (h-index 296) is well-summarized in the paper abstract:

> "*This article explores how agile practices can reduce three kinds of distance temporal, geographical, and sociocultural in global software development (GSD). On the basis of two in-depth case studies, specific Scrum and eXtreme Programming (XP) practices are found to be useful for reducing communication, coordination, and control problems that have been associated with GSD.*"

Communication is a central issue in GSE and the paper "Essential communication practices for Extreme Programming in a global software development team" [LWDB06] (h-index 293) proposes four key factors - team, procsss, project and outcome - for successful communication management in globally distributed agile development (using XP). The study is based on collection of both quantitative and qualitative data and application of grounded theory. they conclude that:

> "*if these critical enabling factors are addressed, methodologies dependent on informal communication can be used on global software development projects.*"

A highly influential set of guidelines for GSE is published in the "Global software development handbook" [SBM+06] (h-index 269). This handbook is based on 2 years experience of GSE projects in a multinational software company, and has high educational value:

> "*This book is for both software engineering practitioners and students of project management focused on distributed development of software. It should be particularly useful to project managers who are planning a distributed project, but those currently involved with such projects may also find useful advice and a fresh perspective on this subject. It can be used as a supplement to an undergraduate- or graduate-level course in software project management, especially when discussing issues surrounding globally distributed software development.*"

"A process framework for global software engineering teams" [RCM+12] (h-index 119) is motivated by the new challenges posed by GSE to the software process:

> "*de facto process frameworks such as the Capability Maturity Model Integration (CMMI) do not explicitly cater for the complex and changing needs of global software management.*"

The paper focuses on 2 research questions:

- What are the threats faced by global software organizations if they do not implement GSE processes correctly?
- Can the CMMI software process improvement structure be used as a basis for developing a process area to support team management in a global software engineering environment?

The paper provides concrete answers to the first question, emphasising the need for companies to understand the risks associated with GSE. The second question is much more challenging and the paper propses only that this may be feasible and desirable, and that further work is required.

The importance of risk management in GSE is the focus of the paper "Managing risks in global software engineering: principles and practices" [EMJ08] (h-index 72):

> “*Globally distributed software development poses substantial risks to project and product management. Not all eventualities can however be buffered, because in the global economy, developing and implementing products must be fast, cost effective and adaptive to changing needs. Therefore, there is a need to utilize different techniques to effectively and efficiently mitigate risks. This article systematically introduces risk management in global software en- gineering (GSE) for product development, service and maintenance.*”

“Patterns in effective distributed software development”[PAS10] (h-index 38) focuses on offshore outsourcing and internal offshoring as 2 of the most important GSE project patterns. The paper provides a literature review of studies into these 2 patterns and emphasises the need for a GSE process model specific to each of the patterns, and which is built upon existing standard models such as the CMMI.

In the paper “The context of global software development: challenges, best practices and benefits”[uHT11] (h-index 21), the authors present a literature review of GSE, and highlight in their conclusions that:

> “*much of the research in this area has been focused on addressing issues faced by client organizations, however, vendor side in the GSD relationship is much ignored due to which this field of study is still immature.*”

The final publication in this section comes back to the question of global agile, with a focus on team configuration — “Best practices for configuring globally distributed agile teams”[SR11](h-index 8). It makes 8 concrete recommendations, based on analysis of a large number of case studies:

- Increase task meaningfulness via agile practices like short iterations and small releases
- Provide a high to moderate degree of autonomy
- Emphasize regular feedback via agile practices such as daily stand-up meetings, iteration planning, iteration demos, and iteration retrospectives, short iterations, small releases, continuous integration, and frequent builds
- Establish expectations and roles upfront, but allow core norms to develop naturally among the team itself
- Keep teams as small as possible or break existing large teams into smaller sub-teams
- Select sites with at least some degree of overlapping work hours not simply the low cost location
- Utilize multiple ICT with an emphasis on teleconferencing, instant messaging, and desktop sharing
- Acknowledge that crossing multiple boundaries does not have to negatively impact the team, but upfront preparation is vital

Global Software Engineering Best Practices Top 10 Ranking

[1] Erran Carmel and Ritu Agarwal. Tactical approaches for alleviating distance in global software development. *IEEE software*, 18(2):22–29, 2001. (h-index 903).

[2] James D Herbsleb, Daniel J Paulish, and Matthew Bass. Global software development at siemens: experience from nine projects. In *Proceedings of the 27th international conference on Software engineering*, pages 524–533. ACM, 2005. (h-index 303).

[3] Helena Holmström, Brian Fitzgerald, Pär J Ågerfalk, and Eoin Ó Conchúir. Agile practices reduce distance in global software development. *Information systems management*, 23(3):7–18, 2006. (h-index 296).

[4] Lucas Layman, Laurie Williams, Daniela Damian, and Hynek Bures. Essential communication practices for extreme programming in a global software development team. *Information and software technology*, 48(9):781–794, 2006. (h-index 293).

[5] Raghvinder Sangwan, Matthew Bass, Neel Mullick, Daniel J Paulish, and Juergen Kazmeier. *Global software development handbook*. Auerbach Publications, 2006. (h-index 269).

[6] Ita Richardson, Valentine Casey, Fergal Mccaffery, John Burton, and Sarah Beecham. A process framework for global software engineering teams. *Information and Software Technology*, 54(11):1175–1191, 2012. (h-index 119).

[7] Christof Ebert, Bvs Krishna Murthy, and Namo Narayan Jha. Managing risks in global software engineering: principles and practices. In *2008 IEEE International Conference on Global Software Engineering*, pages 131–140. IEEE, 2008. (h-index 72).

[8] Rafael Prikladnicki, Jorge Luis Nicolas Audy, and Forrest Shull. Patterns in effective distributed software development. *IEEE software*, 27(2):12–15, 2010. (h-index 38).

[9] Sami ul Haq and Muhammad Tariq. The context of global software development: challenges, best practices and benefits. *Information Management and Business Review*, 3(4):193–197, 2011. (h-index 21).

[10] Jason H Sharp and Sherry D Ryan. Best practices for configuring globally distributed agile teams. *Journal of Information Technology Management*, 22(4):56, 2011. (h-index 8).

3.15 Global Labs Best Practices

This section reviews some of the most influential publications on global education using remote (virtual) laboratories.

One of the earliest use of global labs is reported in "Remote laboratories versus virtual and real laboratories" [NMN03] (h-index 397). The paper compares virtual labs (and simulations) with remote laboratories. The need for good learner interface design is emphasised. They conclude with a recommendation for a blended approach:

> *"We believe that engineering students should be offered through the duration of their programs a balanced mixture of real, virtual and remote labs."*

The advantages of on-line labs are discussed in "Virtual laboratories in engineering education: The simulation lab and remote lab"[BW09] (h-index 323):

> "*Online labs offer vast advantages in engineering laboratory education and become an alternative to physical labs. The simulation lab is a simplified version of a system, and though much criticized for not giving sufficient real experiences, there are a several advantages over real labs, such as flexibility, explanation of theoretical concept, and repetition. The remote lab allows users to control and perform experiments on real equipments via Internet. The benefits of the remote lab are a mixture real and simulation lab advantages. The effectiveness of the remote lab depends on the user interactivity.*"

An excellent recent literature review of global labs is found in "Virtual laboratories for education in science, technology, and engineering: A review"[PGC+16] (h-index 281). The review focuses on technological rather than pedagogic issues:

> "*besides the technical issues addressed in this paper, the implementation of immersive education, distance learning, and virtual worlds open up significant questions in the field of pedagogy and the design of effective learning experiences.*"

Empirical research is important when judging the effectiveness of global labs in teaching. The paper "Constructing reality: A study of remote, hands-on, and simulated laboratories" [CNE+07] (h-index 260) "describes the results of a large-scale study comparing learning outcomes and student preferences for several different lab formats in an undergraduate engineering course". Their results support the use of global labs:

> "*Our results indicate that remote and simulated labs can be at least as effective as traditional hands-on labs in teaching specific course concepts. Students do express preference for traditional hands-on labs, but learn the relevant concepts as well or better with the newer forms of laboratories.*"

"A Web-based distributed virtual educational laboratory"[BBF+00] (h-index 240) is one of the first publications reporting on the use of web architecture for implementation and deployment of a global lab. The system is well-specified, covering educational goals; user accessibility; cost limitation and resource optimisation; software cost and sharing; real-time operation; distributed system best practices; cooperative development, management and maintenance; and security. Their experimental platform has shown that the proposed environment is effective for the two main goals: cost reduction and students' satisfaction.

"Using remote laboratories to extend access to science and engineering"[CSC02] (h-index 119) reports on the EU funded project PEARL - Practical Experimentation by Accessible Remote Learning:

> "*The PEARL system extends Internet course delivery to accommodate collaborative working in practical experimentation by developing and integrating a collaborative working environment with an accessible user interface and a modular system for flexibly creating remotely controlled experiments.*"

The project demonstrates an internet-based proof-of-concept prototype. A follow-up paper validates the PEARL architecture and pedagoic approach: "Remote experiments, re-versioning and re-thinking science learning"[SCCDP04] (h-index 118). The paper concludes with interesting observations concerning disabled students:

"The students with disabilities who used the PEARL system found significant improvements in their experience of laboratory work in the remote setting. However, these students were also concerned that existing provision for disabled students at residential schools should not be reduced."

The need to coherently global labs in learning management systems is discussed in "Remote labs as learning services in the educational arena"[SCM+11] (h-index 43). The development of such a LMS service to integrate web-based and remote labs is the main contribution of the paper.

One of the first publications on the use of global labs comes from the mid 1980s: "The world as an international science laboratory: Electronic networks for science instruction and problem solving"[Lev85] (h-index 41). This paper is extremely insightful and correctly predicts the enormous impact of the internet on teaching.

The final publication in this section proposes hybrid global labs which provide an overlap between virtual and remote teaching laboratories — "Virtual labs and remote labs: practical experience for everyone"[FKPK14] (h-index 17). The paper warns of the the additional development and maintenace costs for having a coherently integrated virtual education platform.

Global Labs Best Practices Top 10 Ranking

[1] Zorica Nedic, Jan Machotka, and Andrew Nafalski. *Remote laboratories versus virtual and real laboratories*, volume 1. IEEE, 2003. (h-index 397).

[2] Balakrishnan Balamuralithara and Peter Charles Woods. Virtual laboratories in engineering education: The simulation lab and remote lab. *Computer Applications in Engineering Education*, 17(1):108–118, 2009. (h-index 323).

[3] Veljko Potkonjak, Michael Gardner, Victor Callaghan, Pasi Mattila, Christian Guetl, Vladimir M Petrović, and Kosta Jovanović. Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95:309–327, 2016. (h-index 281).

[4] James E Corter, Jeffrey V Nickerson, Sven K Esche, Constantin Chassapis, Seongah Im, and Jing Ma. Constructing reality: A study of remote, hands-on, and simulated laboratories. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(2):7, 2007. (h-index 260).

[5] Luigino Benetazzo, Matteo Bertocco, Franco Ferraris, Alessandro Ferrero, Carlo Offelli, Marco Parvis, and Vincenzo Piuri. A web-based distributed virtual educational laboratory. *IEEE Transactions on Instrumentation and Measurement*, 49(2):349–356, 2000. (h-index 240).

[6] Chetz Colwell, Eileen Scanlon, and Martyn Cooper. Using remote laboratories to extend access to science and engineering. *Computers & Education*, 38(1-3):65–76, 2002. (h-index 119).

[7] Eileen Scanlon, Chetz Colwell, Martyn Cooper, and Terry Di Paolo. Remote experiments, re-versioning and re-thinking science learning. *Computers & Education*, 43(1-2):153–163, 2004. (h-index 118).

[8] Elio Sancristobal, Manuel Castro, Sergio Martin, Mohamed Tawkif, Alberto Pesquera, Rosario Gil, Gabriel Díaz, and Juan Peire. Remote labs as learning services in the educational arena. In *2011 IEEE Global Engineering Education Conference (EDUCON)*, pages 1189–1194. IEEE, 2011. (h-index 43).

[9] JA Levin. The world as an international science laboratory: Electronic networks for science instruction and problem solving. *Journal of Computers in Mathematics and Science Teaching*, 4:33–35, 1985. (h-index 41).

[10] Sulamith Frerich, Daniel Kruse, Marcus Petermann, and Andreas Kilzer. Virtual labs and remote labs: practical experience for everyone. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, pages 312–314. IEEE, 2014. (h-index 17).

3.16 Global Software Engineering Education Best Practices

The paper "Instructional design and assessment strategies for teaching global software development: a framework" [DHAA06] (h-index 108) is an experience report from a GSE module taught internationally (between institutes in Canada, Australia and Italy). The paper focuses on 3 main issues- – design, teaching and evaluation — in order to provide

> "*a framework of strategies for teaching and assessment of learning of GSD skills [...] we highlighted findings with respect to two dimensions in the framework: the learning of skills of international teamwork and use of CMC tools for remote communication, demonstrating that these learning outcomes had been met.*"

A very comprehensive literature review of GSE education is published in "Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review" [GPO16] (h-index 77). This paper concludes with an extensive structured list of best practices (BPs) in the following categories:

- Knowledge management - 14 BPs
- Engagement and commitment - 14 BPs
- Understanding industrial needs, challenges, goals and problems - 11 BPs
- Providing industrial solutions - 4 BPs
- Mutual respect and trust - 5 BPs
- Be Agile - 2 BPs
- Team work and configuration - 5 BPs
- Researcher On-Site Involvement - 5 BPs
- Research and data collection methods - 20 BPs
- Managing funding, resources, contracts, privacy, ownership - 13 BPs
- Context, constraints and language - 4 BPs

- Research project management - 9 NPs
- Conduct assessment - 7 BPs
- Test/pilot procedures - 5 BPs
- Tool support - 3 BPs

This highly influential paper is highly recommended to all academics considering teaching global software engineering.

"Runestone, an international student collaboration project" [DPA+98] (h-index 62) reports on one of the earliest educational collaborations on GSE. It involved a European (Swedish) institute and a N. American (Michigan) institute. The project objective was to add:

> "*new dimensions to student teamwork, requiring students to handle collaboration that is remote, cross-cultural, and linguistically challenging.*"

The project encouraged peer teaching/learning. The primary research question was whether a global project was a good mechanism for teaching about GSE. They concluded that the project was a qualfiied success. From the students' perspective, there was a high degree of satisfaction from working on a real project, combined with frustration associated with (international) group interaction. For the academics, it was clear that the teaching staff need to be experienced in GSE as well as experienced teachers.

A Systematic Literature Review (SLR) in the field of GSD training and education is published in "Preparing students and engineers for global software development: a systematic review" [MVPC10] (h-index 61), where the main findings help researchers, instructors and practitioners to discover the main challenges, strategies and recent proposals. The paper concludes with six valuable observations:

- "There is a growing interest in GSD training and education, since GSD has grown in recent years and training has become essential."
- "The teaching and training of GSD must be supported by practical experiences through which students can learn by doing."
- "Simulating the complexity of real environments is difficult for universities, and the different timetables of the students make it difficult to coordinate training projects."
- "It is not possible for instructors to cover all the stages and problems of GSD, so any initiative should be focused on a specific field."
- "Students involved in GSD training programs usually experience a lack of motivation, schedule problems and communication difficulties, and this is greater still when cultural and language differences are present."
- "Particular training scenarios and learning environments require specific tools for communication, collaboration and document management."

"Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review" [CBB+15] (h-index 41) aims to

> "*ameliorate the very difficult task of teaching GSE by delineating the challenges and providing some recommendations for overcoming them.*"

They analysed 82 previously published papers that addressed the main aim, and provided a structured guide to best practice:

> "*We identified seven major themes around which challenges and recommendations were grouped, namely: Global Distance; Teamwork; Curriculum/Pedagogy; Stakeholder/role; Infrastructure; People/Soft issues; Development Process. A comprehensive and detailed set of challenges and associated recommendations have been outlined in this report.*"

"Teaching practical software engineering and global software engineering: case study and recommendations" [PTT06b] (h-index 34) is an interesting experience report of international collaboration between Europe and N. America. Their approach was:

> "*to combine and synchronize class teaching about SW engineering methods and processes with actual SW development work in a setting designed to simulate a small SW company.*"

The paper concludes with 9 concrete recommendations

- "The combination of teaching of SE processes and methods with significant student group project exercises, intensive instructor interaction, and simulation of full SW lifecycle is critical. The synchronization of class material with final project milestones is very beneficial to students"
- 'Students should be formed into small groups of 4-6 members to simulate real life SW companies and go through full SW lifecycle via well-defined milestones. Student groups should be selected by instructors, and have balanced skills, with one student selected as a group lead. Early class surveys can help in such selection."
- "Students should have structured time where they can work in a group setting, preferably immediately after classroom teaching portion of the class."
- "Grading polices should reflect and encourage adherence to SE methods, group interaction and iterations, and not be based solely upon the final delivery."
- " Where possible, implementation or simulation of global SE should be done by having student groups formed of members who do not meet face to face. Care should be taken up front to ensure that local and global groups get to know each other at the beginning and establish an effective teamwork atmosphere."
- " Some class time should be devoted in educating students on the particular SW tools to be used."
- "In the case of global SE, instructor preparation should include close cooperation between instructors of all participating institutions in establishing similar or complementary class goals, material coverage, grading policy etc."
- "It is highly desirable that instructors have real-life experience in delivering SW products and services."

- "Student final projects should be kept live for some time after the class in order to serve as their e-portfolios"

"Working across borders: Overcoming culturally-based technology challenges in student global software development" [GKSN08] (h-index 35) is an interesting experience report concerning collaboration between institutes in N. America (New York) and Asia (India and Cambodia). The paper finds that adoption of appropriate tools is a central issue:

> "*training on tools is necessary but not sufficient to trigger adoption of the tools by students. It must be coupled with evidence of the use of these tools in the industry for students to appreciate their importance.*"

This issue is particularly problematic because different cultures have a different perception on technology-adoption.

The same authors widen their research (including new academic partners) when a novel competition-based approach to GSE education is reported in "A global and competition-based model for fostering technical and soft skills in software engineering education" [GKS+09] (h-index 28), where different teams competed to deliver the best software solution to a real client's needs. The main lesson provided by the paper concerns the need to start small when introducing such a project-based GSE module —

> "*Instructors should not under-estimate the effort and planning required prior to and whilst running such a project, and there are also a number of tasks to attend to on completion. Our recommendation is to start small, cultivate trust amongst instructors, establish supporting relationships amongst students, and then scale.*"

"Ten tips to succeed in global software engineering education" [CBŽ12] (h-index 22) gives a teacher-oriented perspective (based on 9 years experience) on teaching GSE:

> "*Teaching in a distributed environment is, however, very demanding, challenging and unpredictable compared to teaching in a local environment.*"

As the title of the paper suggests, the authors conclude with 10 general tips for the lecturers.

The final paper in this section - "Towards a GSE international teaching network: Mapping Global Software Engineering courses" [FCMP12] (h-index 19) - provides a literature review of experiences in GSE education, and motivates the building of a body of knowledge to help academics in this teaching endeavour. The paper concludes by proposing the construction of a repository for GSE education experience.

Global Software Engineering Education Best Practices Top 10 Ranking

[1] Daniela Damian, Allyson Hadwin, and Ban Al-Ani. Instructional design and assessment strategies for teaching global software development: a framework. In *Proceedings of the 28th international conference on Software engineering*, pages 685–690. ACM, 2006. (h-index 108).

[2] Vahid Garousi, Kai Petersen, and Baris Ozkan. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology*, 79:106–127, 2016. (h-index 77).

[3] Mats Daniels, Marian Petre, Vicki Almstrum, Lars Asplund, C Bjorkman, Carl Erickson, Bruce Klein, and Mary Last. Runestone, an international student collaboration project. In *FIE'98. 28th Annual Frontiers in Education Conference. Moving from'Teacher-Centered'to'Learner-Centered'Education. Conference Proceedings (Cat. No. 98CH36214)*, volume 2, pages 727–732. IEEE, 1998. (h-index 62).

[4] Miguel J Monasor, Aurora Vizcaino, Mario Piattini, and Ismael Caballero. Preparing students and engineers for global software development: a systematic review. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 177–186. IEEE, 2010. (h-index 61).

[5] Tony Clear, Sarah Beecham, John Barr, Mats Daniels, Roger McDermott, Michael Oudshoorn, Airina Savickaite, and John Noll. Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. In *Proceedings of the 2015 ITiCSE on Working Group Reports*, pages 1–39. ACM, 2015. (h-index 41).

[6] Dragutin Petkovic, Rainer Todtenhoefer, and Gary Thompson. Teaching practical software engineering and global software engineering: case study and recommendations. In *Proceedings. Frontiers in Education. 36th Annual Conference*, pages 19–24. IEEE, 2006. (h-index 34).

[7] Olly Gotel, Vidya Kulkarni, Christelle Scharff, and Longchrea Neak. Working across borders: Overcoming culturally-based technology challenges in student global software development. In *2008 21st Conference on Software Engineering Education and Training*, pages 33–40. IEEE, 2008. (h-index 35).

[8] Olly Gotel, Vidya Kulkarni, Moniphal Say, Christelle Scharff, and Thanwadee Sunetnanta. A global and competition-based model for fostering technical and soft skills in software engineering education. In *2009 22nd Conference on Software Engineering Education and Training*, pages 271–278. IEEE, 2009. (h-index 28).

[9] Ivica Crnković, Ivana Bosnić, and Mario Žagar. Ten tips to succeed in global software engineering education. In *Proceedings of the 34th International Conference on Software Engineering*, pages 1225–1234. IEEE Press, 2012. (h-index 22).

[10] Luiz Leandro Fortaleza, Tayana Conte, Sabrina Marczak, and Rafael Prikladnicki. Towards a gse international teaching network: Mapping global software engineering courses. In *Proceedings of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development*, pages 1–5. IEEE Press, 2012. (h-index 19).

3.17 Global Labs for Global Software Engineering Education Best Practices

This final section of the bibliography is the most specialised. By identifying the most influential papers in this area, we were able to seed the literature review search for the most influential papers (on this topic) but from more general areas. As this is the most specific area, and because it is also the most recent in the sense that work in this area is only about a decade old, it is not surprising that the number of citations is lower than in the previous sections. However, each of the following publications have been highly influential in our own work in building a global lab. for GSE education.

The paper "Global software development with cloud platforms" [YRB+09] (h-index 46) would, at first glance, not appear to be education-oriented. However, a significant contribtuon of this paper is the development of a prototype *Lab Any Where(LAW): Online Virtual Labs* to help in the training of software engineers to adopt best practices in GSE. This is the first published work to report on the successful implementation and use of such a platform.

"Glose-lab: Teaching global software engineering." [DHH+11] (h-index 34) presents a teaching environment for global software engineering — the GloSE- Lab. The Glose-lab is a distributed eductional platform that helps co-ordinate software development and educational aspects in a coherent framework. The authors identify communication as the central issue with regards to best practices:

> "*Most of these suggestions relate to communication and coordination between the students, but also between the academic staff. In the end, the organization of a course for global software engineering is also a distributed project.*"

Global labs can take different forms, one of which is the use of simulation of the GSE environment. The paper "Cultural and linguistic problems in GSD: a simulator to train engineers in these issues'[MVP12a] (h-index 22) reports on such a virtual training environment with a focus on cultural and linguistic differences that

> "*can simulate global software development scenarios involving virtual agents (VAs) from different cultures. The VAs interact with learners, who use typical communication tools to solve predefined problems. This environment considers common problems caused by distance and cultural differences when using English as a means of communication. It allows learners to train at any time, because the VAs are always available, and it also permits them to play different roles in the various stages of the project.* "

This paper is complemented by "Providing training in GSD by using a virtual environment." [MVP12b] (h-index 8), in which the authors focus more on the virtual assistant support tool VENTURE (Virtual ENvironment for Training cUlture and language problems in global softwaRe dEvelopment). A novel contribution of this tool is a Virtual Colleague (VC) who will guide the student by giving advice and correcting the inappropriate interactions with the VAs. The platform is yet to be adequately evaluated, but initial feedback from students and staff is promising.

"Project-driven learning-by-doing method for teaching software engineering using virtualization technology'[MTDZ14](h-index 18) describes experiences and lessons on integrating virtualization into senior software engineering course, allowing testing and deploying applications in virtual machines. As well as lectures, the students participated in virtual labs on version control and UML modelling. A major issue was that students had issues in accessing the virtual machines from outside campus — some were unaware of how to use SSH or VNC and many were not aware of how to pause/save virtual machine sessions. Thus, the main lesson to be learned is not to assume that students understand the underlying support technology.

The earliest publication in the use of global labs. and possible best practices was concerned with a very specialised area of software engineering, namely formal methods: "Web-based laboratories in the introductory curriculum enhance formal methods." [JRS96] (h-index 13) The paper offers a few recomendations for best practices: combining formal with fun, provide tools with simple GUIs, continually update and maintain material, support easy and quick experimentation.

The use of virtual meetings as part of practical course work is reported in "Establishing trust and relationships through video conferencing in virtual collaborations: An experience report on a global software engineering course." [HB16] (h-index 6). They observe that trust is fundamental —

> "*both students and practitioners consider trust, goodwill, and a good relationship as important aspects in distributed communication and collaboration. Furthermore, we highlight that participating in even a single virtual meeting can help the students experience some of the challenges in GSE.*"

The use of serious games — as in "GSDgame: A serious game for the acquisition of the competencies needed in GSD." [VVGM+16] (h-index 6) — is an established best practice in teaching GSE, where:

> "*The game simulates scenarios that usually occur in the overall development of a software project, thus enabling the user to become aware of the problems concerning GSD and gain some experience in solving these problems.*"

The game was validated and tested by an expert in serious games by means of a quality model but it has still to be validated in the classroom.

"A virtual, global classroom to teach global software engineering: A Mongolian-German team-teaching project." [ELBA13] (h-index 5) examines 6 challenges in GSE - geographic distance, different time zones, language differences, trust, and cultural differences - within the context of a virtual classroom for project work. An interesting aspect of this paper is the recommendations it makes when the collaborating teams are forced to communicate in a 3rd common language (in this case English is used by Germans and Mongolians).

The final paper in this section — "Deploying an online software engineering education program in a globally distributed organization." [HDM+14] (h-index 4) — is concerned with providing " a set of replicable best practices for initiating a software training program in a multi-national organization". The paper describes "how a web-based communications training platform including e-learning, webinar, video and virtual lab courses was implemented". The virtual global labs that they developed gave learners access to virtual machines and exercises that were followed in an autonomous self-learning manner.

Global Labs for Global Software Engineering Education Best Practices Top 10 Ranking

[1] Pavan Yara, Ramaseshan Ramachandran, Gayathri Balasubramanian, Karthik Muthuswamy, and Divya Chandrasekar. Global software development with cloud platforms. In *International Conference on Software Engineering Approaches for Offshore and Outsourced Development*, pages 81–95. Springer, 2009. (h-index 46).

[2] Constanze Deiters, Christoph Herrmann, Roland Hildebrandt, Eric Knauss, Marco Kuhrmann, Andreas Rausch, Bernhard Rumpe, and Kurt Schneider. Glose-lab: Teaching global software engineering. In *2011 IEEE Sixth International Conference on Global Software Engineering*, pages 156–160. IEEE, 2011. (h-index 34).

[3] Miguel J Monasor, Aurora Vizcaíno, and Mario Piattini. Cultural and linguistic problems in gsd: a simulator to train engineers in these issues. *Journal of Software: Evolution and Process*, 24(6):707–717, 2012. (h-index 22).

[4] Kun Ma, Hao Teng, Lixin Du, and Kun Zhang. Project-driven learning-by-doing method for teaching software engineering using virtualization technology. *International Journal of Emerging Technologies in Learning (iJET)*, 9(9):26–31, 2014. (h-index 18).

[5] Rhys Price Jones, Fritz Ruehr, and Richard Salter. Web-based laboratories in the introductory curriculum enhance formal methods. *ACM SIGCSE Bulletin*, 28(1):160–164, 1996. (h-index 13).

[6] Miguel J Monasor, Aurora Vizcaíno, and Mario Piattini. Providing training in gsd by using a virtual environment. In *International Conference on Product Focused Software Process Improvement*, pages 203–217. Springer, 2012. (h-index 8).

[7] Waqar Hussain and Kelly Blincoe. Establishing trust and relationships through video conferencing in virtual collaborations: An experience report on a global software engineering course. In *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*, pages 49–54. IEEE, 2016. (h-index 6).

[8] David Valencia, Aurora Vizcaino, Lilia Garcia-Mundo, Mario Piattini, and Juan Pablo Soto. Gsdgame: A serious game for the acquisition of the competencies needed in gsd. In *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*, pages 19–24. IEEE, 2016. (h-index 6).

[9] Markus Ende, Ralf Lämmermann, Patricia Brockmann, and Ganchimeg Ayurzana. A virtual, global classroom to teach global software engineering: A mongolian-german team-teaching project. In *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)*, pages 229–233. IEEE, 2013. (h-index 5).

[10] John Hudepohl, Alpana Dubey, Sylvie Moisy, Jessica Thompson, and Hans-Martin Niederer. Deploying an online software engineering education program in a globally distributed organization. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 301–310. ACM, 2014. (h-index 4).

4 Conclusions

In this section, we provide a short summary of its major contributions, insights and guidelines for how the corpus can be used, maintained and extended in the future.

The original research objectives were to attempt to answer the following questions through examination of published literature:

- What are the best practices for GLs?
- What are the best practices for GSE?
- What are the best practices for using GLs for teaching GSE?

We have seen that these very specific questions led — in a bottom-up fashion — to the identification of highly influential publications from wider disciplines. Our objective was not to identify the most influential publications in these wider disciplines, in general, but to identify those that had most influence on the specific questions that we posed.

Our bottom-up approach led us to identify a structured corpus of 17 subcategories of publications: the most specific being directly related to our 3 main questions, the most general being `education`, `global working` and `software engineering`. We argue that this approach is sound - clearly, for example, the best practices in teaching global software engineering will include the best practices in teaching and the best practices in global software engineering.

We note that the most highly cited publications in our corpus come from the most general categories in our corpus structure. The more specific categories correspond to much more recent publications and so it is not surprising that these have, for now, a smaller number of citations.

For completeness, we list the top 10 most influential papers in the whole corpus in the following.

Overall Corpuse Top 10 Ranking

[1] John Seely Brown, Allan Collins, and Paul Duguid. Situated cognition and the culture of learning. *Educational researcher*, 18(1):32–42, 1989. (h-index 20836).

[2] Benjamin Samuel Bloom. Taxonomy of educational objectives: The classification of educational goals. *Cognitive domain*, 1956. (h-index 9501).

[3] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988. (h-index 7462).

[4] Barry W Boehm. A spiral model of software development and enhancement. *Computer*, (5):61–72, 1988. (h-index 6811).

[5] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The unified software development process*, volume 1. Addison-wesley Reading, 1999. (h-index 6428).

[6] Charles C Bonwell and James A Eison. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991. (h-index 6419).

[7] Gilly Salmon. *E-moderating: The key to teaching and learning online*. Psychology Press, 2004. (h-index 5296).

[8] Michael G Moore. Three types of interaction, 1989. (h-index 4428).

[9] Howard S Barrows. A taxonomy of problem-based learning methods. *Medical education*, 20(6):481–486, 1986. (h-index 3254).

[10] Gerry Stahl, Timothy D Koschmann, and Daniel D Suthers. *Computer-supported collaborative learning*. lulu.com, 2006. (h-index 2989).

The complete corpus bibliography can be found in the next section.

References

[AG98] Terry Anderson and D Randy Garrison. Learning in a networked world: New roles and responsibilties. In *Distance Learners in Higher Education: Institutional responses for quality outcomes. Madison, Wi.: Atwood*. Athabasca University Press, 1998. (h-index 500).

[AG14] Yehia Ibrahim Alzoubi and Asif Qumer Gill. Agile global software development communication challenges: A systematic review. In *Proceedings-Pacific Asia Conference on Information Systems, PACIS 2014*, 2014. (h-index 32).

[AM10] Sameer Abufardeh and Kenneth Magel. The impact of global software cultural and linguistic aspects on global software development process (gsd): Issues and challenges. In *4th International Conference on New Trends in Information Science and Service Science*, pages 133–138. IEEE, 2010. (h-index 38).

[And02] Hayward P Andres. A comparison of face-to-face and virtual software development teams. *Team Performance Management: An International Journal*, 8(1/2):39–48, 2002. (h-index 306).

[And04] Terry Anderson. Towards a theory of online learning. In Terry Anderson, editor, *Theory and practice of online learning*, volume 2, pages 109–119. Athabasca University Press, 2004. (h-index 1115).

[Bal14] Maha Bali. Mooc pedagogy: gleaning good practice from existing moocs. *Journal of Online Learning and Teaching*, 10(1):44, 2014. (h-index 257).

[Bar86] Howard S Barrows. A taxonomy of problem-based learning methods. *Medical education*, 20(6):481–486, 1986. (h-index 3254).

[Bat05] AW Tony Bates. *Technology, e-learning and distance education*. Routledge, 2005. (h-index 3447).

[BBF+00] Luigino Benetazzo, Matteo Bertocco, Franco Ferraris, Alessandro Ferrero, Carlo Offelli, Marco Parvis, and Vincenzo Piuri. A web-based distributed virtual educational laboratory. *IEEE Transactions on Instrumentation and Measurement*, 49(2):349–356, 2000. (h-index 240).

[BBH+08] Sarah Beecham, Nathan Baddoo, Tracy Hall, Hugh Robinson, and Helen Sharp. Motivation in software engineering: A systematic literature review. *Information and software technology*, 50(9-10):860–878, 2008. (h-index 494).

[BBVB+01] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001. (h-index 2605).

[BC09] Craig J Bailey and Karen A Card. Effective pedagogical practices for online teaching: Perception of experienced instructors. *The Internet and Higher Education*, 12(3-4):152–155, 2009. (h-index 194).

[BCB+17] Sarah Beecham, Tony Clear, John Barr, Mats Daniels, Michael Oudshoorn, and John Noll. Preparing tomorrow's software engineers for work in a global environment. *IEEE Software*, 34(1):9–12, 2017. (h-index 21).

[BCD89] John Seely Brown, Allan Collins, and Paul Duguid. Situated cognition and the culture of learning. *Educational researcher*, 18(1):32–42, 1989. (h-index 20836).

[BCŽ+10] Ivana Bosnic, Igor Cavrak, Mario Žagar, Rikard Land, and Ivica Crnkovic. Customers' role in teaching distributed software development. In *2010 23rd IEEE Conference on Software Engineering Education and Training*, pages 73–80. IEEE, 2010. (h-index 27).

[BDA+99] P. Bourque, R. Dupuis, A. Abran, J.W. Moore, and L. Tripp. The guide to the software engineering body of knowledge. *Software, IEEE*, 16(6):35–44, 1999. (h-index 627).

[BE91] Charles C Bonwell and James A Eison. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991. (h-index 6419).

[BF04] David Boland and Brian Fitzgerald. Transitioning from a co-located to a globally-distributed software development team: A case study at analog devices inc. In *The 3rd international workshop on global software development*, pages 4–7. IET, 2004. (h-index 56).

[BFMU14] Ricardo Britto, Vitor Freitas, Emilia Mendes, and Muhammad Usman. Effort estimation in global software development: A systematic literature review. In *2014 IEEE 9th International Conference on Global Software Engineering*, pages 135–144. IEEE, 2014. (h-index 38).

[BJ95] Frederick P Brooks Jr. *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E.* Pearson Education India, 1995. (h-index 1025).

[BJMN09] Kay Berkling, Mathai Joseph, Bertrand Meyer, and Martin Nordio, editors. *Software Engineering Approaches for Offshore and Outsourced Development, Second International Conference, SEAFOOD 2008, Zurich, Switzerland, July 2-3, 2008. Revised Papers*, volume 16 of *Lecture Notes in Business Information Processing.* Springer, 2009.

[BKB+07] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4):571–583, 2007.

[BL14] Muhammad Ali Babar and Christian Lescher. Global software engineering: Identifying challenges is important and providing solutions is even better. *Information and Software Technology*, 56(1):1–5, 2014. (h-index 45).

[Blo56] Benjamin Samuel Bloom. Taxonomy of educational objectives: The classification of educational goals. *Cognitive domain*, 1956. (h-index 9501).

[BM02] Erin Bradner and Gloria Mark. Why distance matters: effects on cooperation, persuasion and deception. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 226–235. ACM, 2002. (h-index 305).

[BN06] Ronald J Burke and Eddy Ng. The changing nature of work and organizations: Implications for human resource management. *Human Resource Management Review*, 16(2):86–94, 2006. (h-index 464).

[Boe88] Barry W Boehm. A spiral model of software development and enhancement. *Computer*, (5):61–72, 1988. (h-index 6811).

[Bow01] Beverly L Bower. Distance education: Facing the faculty challenge. *Online Journal of Distance Learning Administration*, 4(2):1–6, 2001. (h-index 470).

[BRN15] Sarah Beecham, Ita Richardson, and John Noll. Assessing the strength of global teaming practices: A pilot study. In *2015 IEEE 10th International Conference on Global Software Engineering*, pages 110–114. IEEE, 2015. (h-index 9).

[Bro00] John Seely Brown. Growing up digital: How the web changes work, education, and the ways people learn. *Change: The Magazine of Higher Learning*, 32(2):11–20, 2000. (h-index 1079).

[BW09] Balakrishnan Balamuralithara and Peter Charles Woods. Virtual laboratories in engineering education: The simulation lab and remote lab. *Computer Applications in Engineering Education*, 17(1):108–118, 2009. (h-index 323).

[CA01] Erran Carmel and Ritu Agarwal. Tactical approaches for alleviating distance in global software development. *IEEE software*, 18(2):22–29, 2001. (h-index 903).

[Cal19] Fabio Calefato, editor. *ICGSE '19: Proceedings of the 14th International Conference on Global Software Engineering*, Piscataway, NJ, USA, 2019. IEEE Press.

[Cas10] Valentine Casey. Imparting the importance of culture to global software development. *ACM inroads*, 1(3):51–57, 2010. (h-index 40).

[CBB+15] Tony Clear, Sarah Beecham, John Barr, Mats Daniels, Roger McDermott, Michael Oudshoorn, Airina Savickaite, and John Noll. Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. In *Proceedings of the 2015 ITiCSE on Working Group Reports*, pages 1–39. ACM, 2015. (h-index 41).

[CBŽ12] Ivica Crnković, Ivana Bosnić, and Mario Žagar. Ten tips to succeed in global software engineering education. In *Proceedings of the 34th International Conference on Software Engineering*, pages 1225–1234. IEEE Press, 2012. (h-index 22).

[CC05] Kajal Claypool and Mark Claypool. Teaching software engineering through game design. In *ACM SIGCSE Bulletin*, volume 37, pages 123–127. ACM, 2005. (h-index 182).

[CH04] Catherine Durnell Cramton and Pamela J Hinds. Subgroup dynamics in internationally distributed teams: Ethnocentrism or cross-national learning? *Research in organizational behavior*, 26:231–263, 2004. (h-index 340).

[CL16] Fabio Calefato and Filippo Lanubile. A hub-and-spoke model for tool integration in distributed development. In *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, pages 129–133. IEEE, 2016. (h-index 6).

[Cle10] Tony Clear. Exploring the notion of cultural fit in global virtual collaborations. *ACM Inroads*, 1(3):58–65, 2010. (h-index 14).

[CLS13] Fabio Calefato, Filippo Lanubile, and Francesco Sportelli. Can social awareness foster trust building in global software teams? In *Proceedings of the 2013 International Workshop on Social Software Engineering*, pages 13–16. ACM, 2013. (h-index 8).

[CNE+07] James E Corter, Jeffrey V Nickerson, Sven K Esche, Constantin Chassapis, Seongah Im, and Jing Ma. Constructing reality: A study of remote, hands-on, and simulated laboratories. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(2):7, 2007. (h-index 260).

[CPW05] David Covington, Donna Petherbridge, and Sarah Egan Warren. Best practices: A triangulated support approach in transitioning faculty to online teaching. *Online Journal of Distance Learning Administration*, 8(1):3–14, 2005. (h-index 109).

[Cra01] Catherine Durnell Cramton. The mutual knowledge problem and its consequences for dispersed collaboration. *Organization science*, 12(3):346–371, 2001. (h-index 2170).

[CS07] Stacey L Connaughton and Marissa Shuffler. Multinational and multicultural distributed teams: A review and future agenda. *Small group research*, 38(3):387–412, 2007. (h-index 267).

[CSC02] Chetz Colwell, Eileen Scanlon, and Martyn Cooper. Using remote laboratories to extend access to science and engineering. *Computers & Education*, 38(1-3):65–76, 2002. (h-index 119).

[CSC+09] Marcelo Cataldo, Charles Shelton, Yongjoon Choi, Yun-Yin Huang, Vytesh Ramesh, Darpan Saini, and Liang-Yun Wang. Camel: A tool for collaborative distributed software design. In *2009 Fourth IEEE International Conference on Global Software Engineering*, pages 83–92. IEEE, 2009. (h-index 41).

[CTD19] Fabio Calefato, Paolo Tell, and Alpana Dubey. Summary of the 14th international conference on global software engineering (icgse). *ACM SIGSOFT Software Engineering Notes*, 44(3):30–33, 2019.

[CW05] Catherine Durnell Cramton and Sheila Simsarian Webber. Relationships among geographic dispersion, team processes, and effectiveness in software development work teams. *Journal of Business Research*, 58(6):758–765, 2005. (h-index 194).

[DAE+05] Clive L Dym, Alice M Agogino, Ozgur Eris, Daniel D Frey, and Larry J Leifer. Engineering design thinking, teaching, and learning. *Journal of engineering education*, 94(1):103–120, 2005. (h-index 2533).

[Dah10] Deepak Dahiya. Teaching software engineering: A practical approach. *ACM SIGSOFT Software Engineering Notes*, 35(2):1–5, 2010. (h-index 23).

[Dam02] Daniela E. Damian. Workshop on global software development. In Will Tracz, Michal Young, and Jeff Magee, editors, *Proceedings of the 24th International Conference on Software Engineering, ICSE 2002, 19-25 May 2002, Orlando, Florida, USA*, pages 667–668. ACM, 2002.

[Dav95] Alan M Davis. *201 principles of software development*. McGraw-Hill, Inc., 1995. (h-index 297).

[DBP99] Mats Daniels, Anders Berglund, and Marian Petre. Reflections on international projects in undergraduate cs education. *Computer Science Education*, 9(3):256–267, 1999. (h-index 33).

[DCPC10] Mats Daniels, Åsa Cajander, Arnold Pears, and Tony Clear. Engineering education research in practice: Evolving use of open ended group projects as a pedagogical strategy for developing skills in global collaboration. *International journal of engineering education*, 26(4):795–806, 2010. (h-index 93).

[DE06] Daniela Damian and Christof Ebert, editors. *1st IEEE International Conference on Global Software Engineering, ICGSE 2006, Florianopolis, Brazil, October 2006*. IEEE Computer Society, 2006.

[Ded96] Chris Dede. The evolution of distance education: Emerging technologies and distributed learning. *American Journal of Distance Education*, 10(2):4–36, 1996. (h-index 844.

[DHAA06] Daniela Damian, Allyson Hadwin, and Ban Al-Ani. Instructional design and assessment strategies for teaching global software development: a framework. In *Proceedings of the 28th international conference on Software engineering*, pages 685–690. ACM, 2006. (h-index 108).

[DHH+11] Constanze Deiters, Christoph Herrmann, Roland Hildebrandt, Eric Knauss, Marco Kuhrmann, Andreas Rausch, Bernhard Rumpe, and Kurt Schneider. Gloselab: Teaching global software engineering. In *2011 IEEE Sixth International Conference on Global Software Engineering*, pages 156–160. IEEE, 2011. (h-index 34).

[DL00] Anke Drappa and Jochen Ludewig. Simulation in software engineering training. In *Proceedings of the 22nd international conference on Software engineering*, pages 199–208. ACM, 2000. (h-index 213).

[DL04] Daniela E. Damian and Filippo Lanubile. The 3rd international workshop on global software development. In Anthony Finkelstein, Jacky Estublier, and David S. Rosenblum, editors, *26th International Conference on Software Engineering (ICSE 2004), 23-28 May 2004, Edinburgh, United Kingdom*, pages 756–757. IEEE Computer Society, 2004.

[DLO03] Daniela Damian, Filippo Lanubile, and Heather L Oppenheimer. Addressing the challenges of software industry globalization: the workshop on global software development. In *Proceedings of the 25th International Conference on Software Engineering*, pages 793–794. IEEE Computer Society, 2003. (h-index 68).

[DLVW99] Soumitra Dutta, Michael Lee, and Luk Van Wassenhove. Software engineering in europe: a study of best practices. *IEEE software*, 16(3):82–90, 1999. (h-index 67).

[DM02] Georgios Dafoulas and Linda Macaulay. Investigating cultural differences in virtual software teams. *The Electronic Journal of Information Systems in Developing Countries*, 7(1):1–14, 2002. (h-index 151).

[DMG07] Paul M Duvall, Steve Matyas, and Andrew Glover. *Continuous integration: improving software quality and reducing risk*. Pearson Education, 2007. (h-index 778).

[DPA+98] Mats Daniels, Marian Petre, Vicki Almstrum, Lars Asplund, C Bjorkman, Carl Erickson, Bruce Klein, and Mary Last. Runestone, an international student collaboration project. In *FIE'98. 28th Annual Frontiers in Education Conference. Moving from'Teacher-Centered'to'Learner-Centered'Education. Conference Proceedings (Cat. No. 98CH36214)*, volume 2, pages 727–732. IEEE, 1998. (h-index 62).

[DSB+09] Georgios A Dafoulas, Kathleen Swigger, Robert Brazile, Ferda Nur Alpaslan, Victor Lopez Cabrera, and Fatma Cemile Serce. Global teams: Futuristic models of collaborative work for today's software development industry. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2009. (h-index 24).

[dSCFP10] Fabio QB da Silva, Catarina Costa, A Cesar C Franca, and Rafael Prikladinicki. Challenges and solutions in distributed software development project management: A systematic literature review. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 87–96. IEEE, 2010. (h-index 134).

[DTMS97] Alan Dutson, Robert H. Todd, Spencer P. Magleby, and Carl D. Sorensen. A review of literature on teaching engineering design through project-oriented capstone courses. *Journal of Engineering Education*, 86:17–28, 1997. (h-index 715).

[Dus02] Elfriede Dustin. *Effective Software Testing: 50 Ways to Improve Your Software Testing*. Addison-Wesley Longman Publishing Co., Inc., 2002. (h-index 215).

[DZ03] Daniela E Damian and Didar Zowghi. RE challenges in multi-site software development organisations. *Requirements engineering*, 8(3):149–160, 2003. (h-index 363).

[EC03] J Alberto Espinosa and Erran Carmel. The impact of time separation on coordination in global software teams: a conceptual foundation. *Software Process: Improvement and Practice*, 8(4):249–266, 2003. (h-index 195).

[Eck13] Jutta Eckstein. *Agile software development with distributed teams: Staying agile in a global world*. Addison-Wesley, 2013. (h-index 47).

[Edm06] Andrea Edmundson. *Globalized e-learning cultural challenges*. IGI Global, 2006. (h-index 164).

[ELBA13] Markus Ende, Ralf Lämmermann, Patricia Brockmann, and Ganchimeg Ayurzana. A virtual, global classroom to teach global software engineering: A mongolian-german team-teaching project. In *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)*, pages 229–233. IEEE, 2013. (h-index 5).

[Ell08] Heidi JC Ellis. *Software Engineering: Effective Teaching and Learning Approaches and Practices: Effective Teaching and Learning Approaches and Practices*. IGI Global, 2008. (h-index 20).

[EMJ08] Christof Ebert, Bvs Krishna Murthy, and Namo Narayan Jha. Managing risks in global software engineering: principles and practices. In *2008 IEEE International Conference on Global Software Engineering*, pages 131–140. IEEE, 2008. (h-index 72).

[ENF$^+$14] Hans-Christian Estler, Martin Nordio, Carlo A Furia, Bertrand Meyer, and Johannes Schneider. Agile vs. structured distributed software development: A case study. *Empirical Software Engineering*, 19(5):1197–1224, 2014. (h-index 101).

[ES03] H Keith Edwards and Varadharajan Sridhar. Analysis of the effectiveness of global virtual teams in software engineering projects. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pages 9–pp. IEEE, 2003. (h-index 111).

[FB04] Richard M Felder and Rebecca Brent. The abcs of engineering education: Abet, blooms taxonomy, cooperative learning, and so on. In *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*, volume 1, 2004. (h-index 191).

[FCMP12] Luiz Leandro Fortaleza, Tayana Conte, Sabrina Marczak, and Rafael Prikladnicki. Towards a gse international teaching network: Mapping global software engineering courses. In *Proceedings of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development*, pages 1–5. IEEE Press, 2012. (h-index 19).

[FD04] Jon Froehlich and Paul Dourish. Unifying artifacts and activities in a visual tool for distributed software development teams. In *Proceedings. 26th International Conference on Software Engineering*, pages 387–396. IEEE, 2004. (h-index 241).

[FG89] Gary A. Ford and Norman E. Gibbs. A master of software engineering curriculum: Recommendations from the software engineering institute. *Computer*, 22:59–71, September 1989. (h-index 69).

[FKPK14] Sulamith Frerich, Daniel Kruse, Marcus Petermann, and Andreas Kilzer. Virtual labs and remote labs: practical experience for everyone. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, pages 312–314. IEEE, 2014. (h-index 17).

[FS+88] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988. (h-index 7462).

[Gar08] K.A. Gary. The software enterprise: Practicing best practices in software engineering education. *International Journal of Engineering Education*, 24(4):705, 2008. (h-index 17).

[GBH08] Gerald Gannod, Janet Burge, and Michael Helmick. Using the inverted classroom to teach software engineering. In *2008 ACM/IEEE 30th International Conference on Software Engineering*, pages 777–786. IEEE, 2008. (h-index 444).

[GC03] Cristina B Gibson and Susan G Cohen. *Virtual teams that work: Creating conditions for virtual team effectiveness*. John Wiley & Sons, 2003. h-index 904.

[GJM02] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of software engineering*. Prentice Hall PTR, 2002. (h-index 1804).

[GJM09] Olly Gotel, Mathai Joseph, and Bertrand Meyer, editors. *Software Engineering Approaches for Offshore and Outsourced Development, Third International Conference, SEAFOOD 2009, Zurich, Switzerland, July 2-3, 2009. Proceedings*, volume 35 of *Lecture Notes in Business Information Processing*. Springer, 2009.

[GKS+09] Olly Gotel, Vidya Kulkarni, Moniphal Say, Christelle Scharff, and Thanwadee Sunetnanta. A global and competition-based model for fostering technical and soft skills in software engineering education. In *2009 22nd Conference on Software Engineering Education and Training*, pages 271–278. IEEE, 2009. (h-index 28).

[GKSN08] Olly Gotel, Vidya Kulkarni, Christelle Scharff, and Longchrea Neak. Working across borders: Overcoming culturally-based technology challenges in student global software development. In *2008 21st Conference on Software Engineering Education and Training*, pages 33–40. IEEE, 2008. (h-index 35).

[GMJY+15] Lucy L Gilson, M Travis Maynard, Nicole C Jones Young, Matti Vartiainen, and Marko Hakonen. Virtual teams research: 10 years, 10 themes, and 10 opportunities. *Journal of management*, 41(5):1313–1337, 2015. (h-index 498).

[GPO16] Vahid Garousi, Kai Petersen, and Baris Ozkan. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology*, 79:106–127, 2016. (h-index 77).

[Gra06] Charles R Graham. Blended learning systems. *The handbook of blended learning*, pages 3–21, 2006. (h-index 2510).

[GS87] David R Garrison and Doug Shale. Mapping the boundaries of distance education: Problems in defining the field. *American Journal of Distance Education*, 1(1):7–13, 1987. (h-index 406).

[Häk04] Päivi Häkkinen. What makes learning and understanding in virtual teams so difficult? *CyberPsychology & behavior*, 7(2):201–206, 2004. (h-index 72).

[HB03] Pamela J Hinds and Diane E Bailey. Out of sight, out of sync: Understanding conflict in distributed teams. *Organization science*, 14(6):615–632, 2003. (h-index 1059).

[HB16] Waqar Hussain and Kelly Blincoe. Establishing trust and relationships through video conferencing in virtual collaborations: An experience report on a global software engineering course. In *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*, pages 49–54. IEEE, 2016. (h-index 6).

[HBP09] Emam Hossain, Muhammad Ali Babar, and Hye-young Paik. Using scrum in global software development: a systematic literature review. In *2009 Fourth IEEE International Conference on Global Software Engineering*, pages 175–184. IEEE, 2009. (h-index 335).

[HCAF06] Helena Holmstrom, Eoin Ó Conchúir, J Agerfalk, and Brian Fitzgerald. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, pages 3–11. IEEE, 2006. (h-index 326).

[HCCK15] Neal Robert Haddaway, Alexandra Mary Collins, Deborah Coughlin, and Stuart Kirk. The role of google scholar in evidence reviews and its applicability to grey literature searching. *PloS one*, 10(9):e0138237, 2015.

[HDM+14] John Hudepohl, Alpana Dubey, Sylvie Moisy, Jessica Thompson, and Hans-Martin Niederer. Deploying an online software engineering education program in a globally distributed organization. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 301–310. ACM, 2014. (h-index 4).

[Her07] James D Herbsleb. Global software engineering: The future of socio-technical coordination. In *Future of Software Engineering (FOSE'07)*, pages 188–198. IEEE, 2007. (h-index 723).

[HFÅC06] Helena Holmström, Brian Fitzgerald, Pär J Ågerfalk, and Eoin Ó Conchúir. Agile practices reduce distance in global software development. *Information systems management*, 23(3):7–18, 2006. (h-index 296).

[HGK05] Guido Hertel, Susanne Geister, and Udo Konradt. Managing virtual teams: A review of current empirical research. *Human resource management review*, 15(1):69–95, 2005. (h-index 1242).

[HM01] James D Herbsleb and Deependra Moitra. Global software development. *IEEE software*, 18(2):16–20, 2001. (h-index 1278).

[Hol01] Judith A Holton. Building trust and collaboration in a virtual team. *Team performance management: an international journal*, 7(3/4):36–47, 2001. h-index 390.

[HP05] Matthew J Hawthorne and Dewayne E Perry. Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities. In *International Conference on Software Engineering*, pages 166–185. Springer, 2005. (h-index 84).

[HPB05] James D Herbsleb, Daniel J Paulish, and Matthew Bass. Global software development at siemens: experience from nine projects. In *Proceedings of the 27th international conference on Software engineering*, pages 524–533. ACM, 2005. (h-index 303).

[HŠM11] Geir K Hanssen, Darja Šmite, and Nils Brede Moe. Signs of agile trends in global software engineering research: A tertiary study. In *2011 IEEE Sixth International Conference on Global Software Engineering Workshop*, pages 17–23. IEEE, 2011. (h-index 66).

[Hua02] Hsiu-Mei Huang. Toward constructivism for adult learners in online learning environments. *British journal of educational technology*, 33(1):27–37, 2002. (h-index 949.

[Hum95] Watts S Humphrey. *A discipline for software engineering*. Addison-Wesley Longman Publishing Co., Inc., 1995. (h-index 2027).

[JBR99] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The unified software development process*, volume 1. Addison-wesley Reading, 1999. (h-index 6428).

[JDC+95] David Jonassen, Mark Davidson, Mauri Collins, John Campbell, and Brenda Bannan Haag. Constructivism and computer-mediated communication in distance education. *American journal of distance education*, 9(2):7–26, 1995. (h-index 1784).

[Jon09] Capers Jones. *Software engineering best practices*. McGraw-Hill, Inc., 2009. (h-index 216).

[JPV09] Miguel Jiménez, Mario Piattini, and Aurora Vizcaíno. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, 2009:3, 2009. (h-index 258).

[JRS96] Rhys Price Jones, Fritz Ruehr, and Richard Salter. Web-based laboratories in the introductory curriculum enhance formal methods. *ACM SIGCSE Bulletin*, 28(1):160–164, 1996. (h-index 13).

[JW10] Samireh Jalali and Claes Wohlin. Agile practices in global software engineering-a systematic map. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 45–54. IEEE, 2010. (h-index 164).

[Kar99] Dale Walter Karolak. *Global software development: managing virtual teams and environments*. IEEE Computer society press, 1999. (h-index 322).

[KK10] Jared Keengwe and Terry T Kidd. Towards best practices in online learning and teaching in higher education. *MERLOT Journal of Online Learning and Teaching*, 6(2):533–541, 2010. (h-index 275).

[KM01] Bruce Kogut and Anca Metiu. Open-source software development and distributed innovation. *Oxford review of economic policy*, 17(2):248–264, 2001. (h-index 696).

[KRG+02] Bradley L Kirkman, Benson Rosen, Cristina B Gibson, Paul E Tesluk, and Simon O McPherson. Five challenges to virtual team success: Lessons from sabre, inc. *Academy of Management Perspectives*, 16(3):67–79, 2002. h-index 713.

[KSW04] Srinivas Krishna, Sundeep Sahay, and Geoff Walsham. Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4):62–66, 2004. (h-index 678).

[LCE13] Filippo Lanubile, Fabio Calefato, and Christof Ebert. Group awareness in global software engineering. *IEEE software*, 30(2):18–23, 2013. (h-index 26).

[LDHRJ+07] Timothy C Lethbridge, Jorge Diaz-Herrera, J Richard Jr, J Barrie Thompson, et al. Improving software practice through education: Challenges and future trends. In *Future of Software Engineering (FOSE'07)*, pages 12–28. IEEE, 2007. (h-index 156).

[Lef07] Dean Leffingwell. *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007. (h-index 450).

[LEPV10] Filippo Lanubile, Christof Ebert, Rafael Prikladnicki, and Aurora Vizcaíno. Collaboration tools for global software engineering. *IEEE software*, 27(2):52–55, 2010. (h-index 211).

[Let98] Timothy C Lethbridge. The relevance of software education: A survey and some recommendations. *Annals of Software Engineering*, 6(1-4):91–110, 1998. (h-index 40).

[Let00] Timothy C Lethbridge. What knowledge is important to a software professional? *Computer*, 33(5):44–50, 2000. (h-index 334).

[Lev85] JA Levin. The world as an international science laboratory: Electronic networks for science instruction and problem solving. *Journal of Computers in Mathematics and Science Teaching*, 4:33–35, 1985. (h-index 41).

[LI08] Chi-Sing Li and Beverly Irby. An overview of online education: Attractiveness, benefits, challenges, concerns and recommendations. *College Student Journal*, 42(2), 2008. (h-index 231).

[LLJS+06] Timothy C Lethbridge, Richard J LeBlanc Jr, Ann E Kelley Sobel, Thomas B Hilburn, and Jorge L Diaz-Herrera. Se2004: Recommendations for undergraduate software engineering curricula. *IEEE software*, 23(6):19–25, 2006. (h-index 54).

[LR01] Jeremy S Lurey and Mahesh S Raisinghani. An empirical study of best practices in virtual teams. *Information & Management*, 38(8):523–544, 2001. h-index 701.

[LRMC87] James A Levin, Margaret Riel, Naomi Miyake, and Moshe Cohen. Education on the electronic frontier: Teleapprentices in globally distributed educational contexts. *Contemporary Educational Psychology*, 12(3):254–260, 1987. (h-index 112).

[LWBC94] James Levin, Michael Waugh, David Brown, and Rene Clift. Teaching teleapprenticeships: A new organizational framework for improving teacher education using electronic networks. *Journal of Machine-Mediated Learning*, 4(2):149–161, 1994. (h-index 52).

[LWDB06] Lucas Layman, Laurie Williams, Daniela Damian, and Hynek Bures. Essential communication practices for extreme programming in a global software development team. *Information and software technology*, 48(9):781–794, 2006. (h-index 293).

[Mar07] Harriet Marshall. The global education terminology debate: Exploring some of the issues. *The SAGE handbook of research in international education*, pages 38–50, 2007. (h-index 84).

[Mar09] Robert Martignoni. Global sourcing of software development-a review of tools and services. In *2009 Fourth IEEE International Conference on Global Software Engineering*, pages 303–308. IEEE, 2009. (h-index 27).

[Mau99] Frank Maurer. ICSE 99 workshop on software engineering over the internet. In Barry W. Boehm, David Garlan, and Jeff Kramer, editors, *Proceedings of the 1999 International Conference on Software Engineering, ICSE' 99, Los Angeles, CA, USA, May 16-22, 1999*, pages 696–697. ACM, 1999.

[Mau00] Frank Maurer. 3rd workshop on software engineering over the internet. In Carlo Ghezzi, Mehdi Jazayeri, and Alexander L. Wolf, editors, *Proceedings of the 22nd International Conference on on Software Engineering, ICSE 2000, Limerick Ireland, June 4-11, 2000*, pages 807–808. ACM, 2000.

[Mau01] Frank Maurer. 4th ICSE workshop on "software engineering over the internet". In Hausi A. Müller, Mary Jean Harrold, and Wilhelm Schäfer, editors, *Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001, 12-19 May 2001, Toronto, Ontario, Canada*, pages 751–752. IEEE Computer Society, 2001.

[MCDE15] Nils Brede Moe, Daniela S Cruzes, Tore Dybå, and Ellen Engebretsen. Coaching a global agile virtual team. In *2015 IEEE 10th International Conference on Global Software Engineering*, pages 33–37. IEEE, 2015. (h-index 26).

[Mey01] Bertrand Meyer. Software engineering in the academy. *Computer*, 34(5):28–35, 2001. (h-index 134).

[MF08] Libby V Morris and Catherine L Finnegan. Best practices in predicting and encouraging student persistence and achievement online. *Journal of College Student Retention: Research, Theory & Practice*, 10(1):55–64, 2008. (h-index 138).

[MH01] Audris Mockus and James Herbsleb. Challenges of global software development. In *Proceedings seventh international software metrics symposium*, pages 182–184. IEEE, 2001. (h-index 176).

[MJ06] Peter Meso and Radhika Jain. Agile software development: adaptive systems principles and best practices. *Information systems management*, 23(3):19–30, 2006. (h-index 203).

[MJ07] Bertrand Meyer and Mathai Joseph, editors. *SEAFOOD'07: Proceedings of the 1st International Conference on Software Engineering Approaches for Offshore and Outsourced Development*, Berlin, Heidelberg, 2007. Springer-Verlag. (h-index 4).

[MK03] Thomas Miller and Frederick King. Distance education: Pedagogy and best practices in the new millennium. *Int. Leadership in Education*, 6(3):283–297, 2003. (h-index 86).

[Moo73] Michael Grahame Moore. Toward a theory of independent learning and teaching. *The Journal of Higher Education*, 44(9):661–679, 1973. (h-index 961).

[Moo89] Michael G Moore. Three types of interaction, 1989. (h-index 4428).

[MTDZ14] Kun Ma, Hao Teng, Lixin Du, and Kun Zhang. Project-driven learning-by-doing method for teaching software engineering using virtualization technology. *International Journal of Emerging Technologies in Learning (iJET)*, 9(9):26–31, 2014. (h-index 18).

[MVP12a] Miguel J Monasor, Aurora Vizcaíno, and Mario Piattini. Cultural and linguistic problems in gsd: a simulator to train engineers in these issues. *Journal of Software: Evolution and Process*, 24(6):707–717, 2012. (h-index 22).

[MVP12b] Miguel J Monasor, Aurora Vizcaíno, and Mario Piattini. Providing training in gsd by using a virtual environment. In *International Conference on Product Focused Software Process Improvement*, pages 203–217. Springer, 2012. (h-index 8).

[MVPC10] Miguel J Monasor, Aurora Vizcaino, Mario Piattini, and Ismael Caballero. Preparing students and engineers for global software development: a systematic review. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 177–186. IEEE, 2010. (h-index 61).

[MW98] David R McIntyre and Francis G Wolff. An experiment with www interactive learning in university education. *Computers & Education*, 31(3):255–264, 1998. (h-index 104).

[NBR10] John Noll, Sarah Beecham, and Ita Richardson. Global software development and collaboration: barriers and solutions. *ACM inroads*, 1(3):66–78, 2010. (h-index 238).

[NDCC15] Anh Nguyen-Duc, Daniela S Cruzes, and Reidar Conradi. The impact of global dispersion on coordination, team performance and software quality–a systematic literature review. *Information and Software Technology*, 57:277–294, 2015. (h-index 101).

[Ng07] Kwok Chi Ng. Replacing face-to-face tutorials by synchronous online technologies: Challenges and pedagogical implications. *The International Review of Research in Open and Distributed Learning*, 8(1), 2007. (h-index 160).

[NJMT10] Martin Nordio, Mathai Joseph, Bertrand Meyer, and Andrey Terekhov, editors. *Software Engineering Approaches for Offshore and Outsourced Development - 4th International Conference, SEAFOOD 2010, St. Petersburg, Russia, June 17-18, 2010. Proceedings*, volume 54 of *Lecture Notes in Business Information Processing*. Springer, 2010.

[NMA+16] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, Mohammed Rehan Riaz, Kanaan Faisal, Narciso Cerpa, Siffat Ullah Khan, and Ita Richardson. Challenges of project management in global software development: A client-vendor analysis. *Information and Software Technology*, 80:1–19, 2016. (h-index 85).

[NMAH15] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, and Ayman Hroub. Empirical investigation of the challenges of the existing tools used in global software development projects. *IET Software*, 9(5):135–143, 2015. (h-index 32).

[NMN03] Zorica Nedic, Jan Machotka, and Andrew Nafalski. *Remote laboratories versus virtual and real laboratories*, volume 1. IEEE, 2003. (h-index 397).

[NPLP12] Tuomas Niinimäki, Arttu Piri, Casper Lassenius, and Maria Paasivaara. Reflecting the choice and usage of communication tools in global software development projects with media synchronicity theory. *Journal of Software: Evolution and Process*, 24(6):677–692, 2012. (h-index 32).

[NRB17] John Noll, Mohammad Abdur Razzak, and Sarah Beecham. Motivation and autonomy in global software development: an empirical study. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 394–399. ACM, 2017. (h-index 9).

[NYAT13] Srinivas Nidhra, Muralidhar Yanamadala, Wasif Afzal, and Richard Torkar. Knowledge transfer challenges and mitigation strategies in global software developmenta systematic literature review and industrial validation. *International journal of information management*, 33(2):333–355, 2013. (h-index 86).

[OBH01] Diana Oblinger, Carole A Barone, and Brian L Hawkins. *Distributed education and its challenges: An overview*, volume 1. American Council on Education Washington, DC, 2001. (h-index 271).

[OC07] Michael Boyer O'Leary and Jonathon N Cummings. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *MIS quarterly*, pages 433–452, 2007. (h-index 663).

[Ols00] J Olson. Distance matters. *Human-Computer Interaction*, 15, 2000. (h-index 2347).

[PA10] Rafael Prikladnicki and Jorge Luis Nicolas Audy. Process models in the practice of distributed software development: A systematic review of the literature. *Information and Software Technology*, 52(8):779–791, 2010. (h-index 84).

[Par99] David Lorge Parnas. Software engineering programs are not computer science programs. *Software, IEEE*, 16(6):19–30, 1999. (h-index 261).

[PAS10] Rafael Prikladnicki, Jorge Luis Nicolas Audy, and Forrest Shull. Patterns in effective distributed software development. *IEEE software*, 27(2):12–15, 2010. (h-index 38).

[PBL+15] Maria Paasivaara, Kelly Blincoe, Casper Lassenius, Daniela Damian, Jyoti Sheoran, Francis Harrison, Prashant Chhabra, Aminah Yussuf, and Veikko Isotalo. Learning global agile software engineering using same-site and cross-site teams. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 285–294. IEEE, 2015. (h-index 21).

[PD99] Guy Paré and Line Dubé. Virtual teams: An exploratory study of key challenges and strategies. *ICIS 1999 Proceedings*, page 50, 1999. h-index 80.

[PDJ09] Rashmi Phalnikar, VS Deshpande, and SD Joshi. Applying agile principles for distributed software development. In *2009 International Conference on Advanced Computer Control*, pages 535–539. IEEE, 2009. (h-index 67).

[Pet71] Otto Peters. Theoretical aspects of correspondence instruction. *McKenzie y Christensen. The changing world of correspondence study. Pensylvania State University Presss*, 1971. (h-index 98).

[PGC+16] Veljko Potkonjak, Michael Gardner, Victor Callaghan, Pasi Mattila, Christian Guetl, Vladimir M Petrović, and Kosta Jovanović. Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95:309–327, 2016. (h-index 281).

[Pig96] Thomas M Pigoski. *Practical software maintenance: best practices for managing your software investment*. Wiley Publishing, 1996. (h-index 714).

[PL06] Maria Paasivaara and Casper Lassenius. Could global software development benefit from agile methods? In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, pages 109–113. IEEE, 2006. (h-index 144).

[PLD+13] Maria Paasivaara, Casper Lassenius, Daniela Damian, Petteri Räty, and Adrian Schröter. Teaching students global software engineering skills using distributed scrum. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1128–1137. IEEE, 2013. (h-index 46).

[PMA12] John Stouby Persson, Lars Mathiassen, and Ivan Aaen. Agile distributed software development: enacting control through media and context. *Information Systems Journal*, 22(6):411–433, 2012. (h-index 83).

[PP99] Rena M Palloff and Keith Pratt. *Building learning communities in cyberspace*, volume 12. San Francisco: Jossey-Bass, 1999. (h-index 4347).

[PP07] Rena M Palloff and Keith Pratt. *Building online learning communities: Effective strategies for the virtual classroom*. John Wiley & Sons, 2007. (h-index 1214).

[PRVEP10] Javier Portillo-Rodriguez, Aurora Vizcaino, Christof Ebert, and Mario Piattini. Tools to support global software development processes: a survey. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 13–22. IEEE, 2010. (h-index 41).

[PRVPB12] Javier Portillo-Rodríguez, Aurora Vizcaíno, Mario Piattini, and Sarah Beecham. Tools used in global software engineering: A systematic mapping review. *Information and Software Technology*, 54(7):663–685, 2012. (h-index 127).

[PTE+11] Jukka-Pekka Pesola, Hannu Tanner, Juho Eskeli, Paivi Parviainen, and Dan Bendas. Integrating early v&v support to a gse tool integration platform. In *2011 IEEE Sixth International Conference on Global Software Engineering Workshop*, pages 95–101. IEEE, 2011. (h-index 11).

[PTT06a] Dragutin Petkovic, Gary Thompson, and Rainer Todtenhoefer. Teaching practical software engineering and global software engineering: evaluation and comparison. *ACM SIGCSE Bulletin*, 38(3):294–298, 2006. (h-index 65).

[PTT06b] Dragutin Petkovic, Rainer Todtenhoefer, and Gary Thompson. Teaching practical software engineering and global software engineering: case study and recommendations. In *Proceedings. Frontiers in Education. 36th Annual Conference*, pages 19–24. IEEE, 2006. (h-index 34).

[QMLB83] Clark N Quinn, Hugh Mehan, James A Levin, and Steven D Black. Real education in non-real time: The use of electronic message systems for instruction. *Instructional Science*, 11(4):313–327, 1983. (h-index 131).

[RCBH11] Narayan Ramasubbu, Marcelo Cataldo, Rajesh Krishna Balan, and James D Herbsleb. Configuring global software teams: a multi-company analysis of project productivity, quality, and profits. In *2011 33rd International Conference on Software Engineering (ICSE)*, pages 261–270. IEEE, 2011. (h-index 85).

[RCM+12] Ita Richardson, Valentine Casey, Fergal Mccaffery, John Burton, and Sarah Beecham. A process framework for global software engineering teams. *Information and Software Technology*, 54(11):1175–1191, 2012. (h-index 119).

[RCMX06] Balasubramaniam Ramesh, Lan Cao, Kannan Mohan, and Peng Xu. Can distributed software development be agile? *Communications of the ACM*, 49(10):41–46, 2006. (h-index 370).

[RdH89] Alexander J Romiszowski and Johan A de Haas. Computer mediated communication for instruction: Using e-mail as a seminar. *Educational Technology*, 29(10):7–14, 1989. (h-index 133).

[Rei06] Thomas J Reichlmay. Collaborating with industry: strategies for an undergraduate software engineering program. In *Proceedings of the 2006 international workshop on Summit on software engineering education*, pages 13–16. ACM, 2006. (h-index 53).

[RKO98] Karen Roberts, Ellen Ernst Kossek, and Cynthia Ozeki. Managing the global workforce: Challenges and strategies. *Academy of Management Perspectives*, 12(4):93–106, 1998. h-index 260.

[RMM+10] Ita Richardson, Sarah Moore, Alan Malone, Valentine Casey, and Dolores Zage. Globalizing software development in the local classroom. In *IT Outsourcing: Concepts, Methodologies, Tools, and Applications*, pages 1534–1556. IGI Global, 2010. (h-index 41).

[RMMK06] Ita Richardson, Allen E Milewski, Neel Mullick, and Patrick Keil. Distributed development: an education perspective on the global studio project. In *Proceedings of the 28th international conference on Software engineering*, pages 679–684. ACM, 2006. (h-index 63).

[Sal04] Gilly Salmon. *E-moderating: The key to teaching and learning online*. Psychology Press, 2004. (h-index 5296).

[SBM+06] Raghvinder Sangwan, Matthew Bass, Neel Mullick, Daniel J Paulish, and Juergen Kazmeier. *Global software development handbook*. Auerbach Publications, 2006. (h-index 269).

[SCCDP04] Eileen Scanlon, Chetz Colwell, Martyn Cooper, and Terry Di Paolo. Remote experiments, re-versioning and re-thinking science learning. *Computers & Education*, 43(1-2):153–163, 2004. (h-index 118).

[Sch98] Lynne Schrum. On-line education: A study of emerging pedagogy. *New directions for adult and continuing education*, 78:53–61, 1998. (h-index 213).

[SCM+11] Elio Sancristobal, Manuel Castro, Sergio Martin, Mohamed Tawkif, Alberto Pesquera, Rosario Gil, Gabriel Díaz, and Juan Peire. Remote labs as learning services in the educational arena. In *2011 IEEE Global Engineering Education Conference (EDUCON)*, pages 1189–1194. IEEE, 2011. (h-index 43).

[Sco00] Peter Scott. Globalisation and higher education: Challenges for the 21st century. *Journal of studies in International Education*, 4(1):3–10, 2000. (h-index 343).

[SH10] Thomas Stober and Uwe Hansmann. *Best Practices for Large Software Development Projects*. Springer, 2010. (h-index 122).

[Sha88] Doug Shale. Concepts: Toward a reconceptualization of distance education. *American journal of distance education*, 2(3):25–35, 1988. (h-index 209).

[Sha00] Mary Shaw. Software engineering education: a roadmap. In *International Conference on Software Engineering — Future of Software Engineering Track*, pages 371–380, 2000. (h-index 257).

[She95] Lorraine Sherry. Issues in distance learning. *International journal of educational telecommunications*, 1(4):337–365, 1995. (h-index 1250).

[SKS06] Gerry Stahl, Timothy D Koschmann, and Daniel D Suthers. *Computer-supported collaborative learning*. lulu.com, 2006. (h-index 2989).

[SM12] R Sriram and SK Mathew. Global software development using agile methodologies: A review of literature. In *2012 IEEE International Conference on Management of Innovation & Technology (ICMIT)*, pages 389–393. IEEE, 2012. (h-index 35).

[Som11] Ian Sommerville. *Software engineering*. Addison-Wesley/Pearson, 2011. (h-index 1768).

[Son96] Ki-Sang Song. Teaching software engineering through real-life projects to bridge school and industry. *ACM SIGCSE Bulletin*, 28(4):59–64, 1996. (h-index 30).

[SR11] Jason H Sharp and Sherry D Ryan. Best practices for configuring globally distributed agile teams. *Journal of Information Technology Management*, 22(4):56, 2011. (h-index 8).

[SS97] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997. (h-index 2149).

[SS09] Saonee Sarker and Suprateek Sarker. Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research*, 20(3):440–461, 2009. (h-index 264).

[SSV04] Carol Saunders, Craig Van Slyke, and Douglas R. Vogel. My Time or Yours? Managing Time Visions in Global Virtual Teams. *The Academy of Management Executive (1993-2005)*, 18(1):19–31, 2004. h-index 253.

[Str87] John Strain. Media and adult learning: A forum: The role of the faculty member in distance education. 1987. (h-index 73).

[SW07] D Sandy Staples and Jane Webster. Exploring traditional and virtual team members best practices a social cognitive theory perspective. *Small group research*, 38(1):60–97, 2007. h-index 123.

[ŠWGF10] Darja Šmite, Claes Wohlin, Tony Gorschek, and Robert Feldt. Empirical evidence in global software engineering: a systematic review. *Empirical software engineering*, 15(1):91–118, 2010. (h-index 361).

[TPBA07] M Rita Thissen, Jean M Page, Madhavi C Bharathi, and Toyia L Austin. Communication tools for distributed software development teams. In *Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce*, pages 28–35. ACM, 2007. (h-index 80).

[uHT11] Sami ul Haq and Muhammad Tariq. The context of global software development: challenges, best practices and benefits. *Information Management and Business Review*, 3(4):193–197, 2011. (h-index 21).

[Var12] Moshe Y Vardi. Will moocs destroy academia? *Communications of the ACM*, 55(11):5–5, 2012. (h-index 284).

[VBK+14] June M Verner, O Pearl Brereton, Barbara A Kitchenham, Mahmood Turner, and Mahmood Niazi. Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, 56(1):54–78, 2014. (h-index 129).

[VDDAS07] Louwarnoud Van Der Duim, Jesper Andersson, and Marco Sinnema. Good practices for educational software engineering projects. In *29th International Conference on Software Engineering (ICSE'07)*, pages 698–707. IEEE, 2007. (h-index 47).

[VdSEPG18] Raoul Vallon, Bernardo Jose da Silva Estacio, Rafael Prikladnicki, and Thomas Grechenig. Systematic literature review on agile practices in global software development. *Information and Software Technology*, 96:161–180, 2018. (h-index 26).

[VGS01] Xavier Vilalta, Denis Gillet, and Christophe Salzmann. Contribution to the definition of best practices for the implementation of remote experimentation solutions. In *IFAC Workshop on Internet Based Control Education (IBCE'01)*, number CONF, 2001. (h-index 14).

[Vin03] Robert Vinaja. Major challenges in multi-cultural virtual teams. *Proceedings: Southwest Case Research Association*, 78541(956):341–346, 2003. h-index 31.

[VL00] Thierry Volery and Deborah Lord. Critical success factors in online education. *International journal of educational management*, 14(5):216–223, 2000. (h-index 988).

[VVGM+16] David Valencia, Aurora Vizcaino, Lilia Garcia-Mundo, Mario Piattini, and Juan Pablo Soto. Gsdgame: A serious game for the acquisition of the competencies needed in gsd. In *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*, pages 19–24. IEEE, 2016. (h-index 6).

[VVVV08] Hans Van Vliet and JC Van Vliet. *Software engineering: principles and practice*, volume 13. Citeseer, 2008. (h-index 1071).

[WN15] Vathsala Wickramasinghe and Sahan Nandula. Diversity in team composition, relationship conflict and team leader support on globally distributed virtual software development team performance. *Strategic Outsourcing: An International Journal*, 8(2/3):138–155, 2015. (h-index 18).

[Woh14] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, page 38. Citeseer, 2014.

[YGDG16] Agustin Yagüe, Juan Garbajosa, Jessica Díaz, and Eloy González. An exploratory study in communication in agile global software development. *Computer Standards & Interfaces*, 48:184–197, 2016. (h-index 30).

[YP13] Li Yuan and SJ Powell. MOOCs and open education: Implications for higher education. *JISC*, 2013. (h-index 788).

[YRB+09] Pavan Yara, Ramaseshan Ramachandran, Gayathri Balasubramanian, Karthik Muthuswamy, and Divya Chandrasekar. Global software development with cloud platforms. In *International Conference on Software Engineering Approaches for Offshore and Outsourced Development*, pages 81–95. Springer, 2009. (h-index 46).

[ZAW04] Norhayati Zakaria, Andrea Amelinckx, and David Wilemon. Working together apart? Building a knowledge-sharing culture for global virtual teams. *Creativity and innovation management*, 13(1):15–29, 2004. (h-index 507).