

Virtual Teams and Employability in Global Software Engineering Education

Julian M. Bass
Robert Gordon University
Aberdeen, UK
Email: j.bass@computer.org

Roger McDermott
Robert Gordon University
Aberdeen, UK

J. T. Lalchandani
IIIT-B
Bangalore, Karnataka, India

Abstract—Universities face many challenges when creating opportunities for student experiences of global software engineering. We provide a model for introducing global software engineering into the computing curriculum. Our model is based on a three year collaboration between Robert Gordon University, UK and the International Institute for IT Bangalore, India. We provide evidence based on student feedback from three cohorts of virtual team who never met face to face. We found potential employers were supportive of global software engineering in university curricula. We identify four key principles for global software engineering student projects: reconcile contrasting assessment demands between institutions, create a detailed joint timetable to reconcile teaching calendars, provide a project management framework to support phased delivery and carefully manage project scope.

I. INTRODUCTION

Worldwide increases in fluidity of information flows and international travel have become known as globalisation [1]. These trends, driven by enhanced network and communication technologies, have triggered changes in software procurement and software engineering processes [2] [3]. The trend towards globalised software production is pioneered by software intensive high-technology businesses [4]. In large enterprises, global software development (GSD) has become the norm involving outsourcing, off-shoring and distributed development models [5].

This paper addresses three related research questions: “how can universities provide opportunities for students to gain first-hand experience of geographically distributed software development projects?”, “What are the perspectives of potential employers on student virtual team projects” and “how can virtual team project experiences be harnessed to improve employability skills in the commercial software sector?”

To address these question we have collected data from students from India and UK participating in international group projects. Three cohorts over a three year period have been successfully conducted with students working in virtual teams, who never met face to face. The three project cohorts evolved in sophistication and scale, reflecting an adoption process for global software engineering in higher education curricula. Further, we have asked business owners and executives to identify the key virtual team working skills they value in graduates.

The paper is structured as follows. First, a review of related literature in global software engineering education,

agile software processes and student project pedagogy is presented. Next, the research methods used in the study are described. Findings from the three cycles of student project and employers are then presented followed by discussion, recommendations and conclusions.

II. GLOBAL SOFTWARE ENGINEERING

Empirical research shows that geographically distributed software development takes longer, and requires more people for work of equal size and complexity when compared with co-located team work [2]. Nevertheless, as has been said, globalisation makes distributed, offshore and outsourced software development a reality. The focus of global software engineering includes geographical distance, temporal distance and socio-cultural distance.

A. Agile Software Development

Agile Software Development is the term used to describe a collection of methods including: Dynamic Systems Development Methods (DSDM) [6], Feature Driven Design [7], Crystal [8], Scrum [9], Extreme Programming [10] and Lean Software Development [11]. Traditionally these approaches were advocated for smaller, co-located development teams. However in recent years, agile development methods have found favour in large scale enterprise offshore and out-sourced software development programmes [12] [13]. The three most important perceived agile principles identified by practitioners are: (1) achievement of customer satisfaction through early and continuous delivery of valuable software, (2) business representatives and development team members working together frequently throughout the project, and (3) face-to-face conversations as the most efficient way to convey information to, and within, the development team [14]. Empirical research reveals that agile methods improve job satisfaction, productivity and customer satisfaction, but with adoption challenges for large-scale projects [15].

B. Global Software Engineering Education

Universities faculty face considerable challenges in teaching knowledge, skills and attitudes in software engineering [16]. In the discipline of computer science, a well established benchmark exists which is the result of efforts over a 40 year period [17]. This benchmark curriculum has influenced curricula development around the world, even in such seemingly remote places as sub-Saharan Africa [18]. The benchmark

identifies 18 topical areas of study in computing including software engineering.

Collaborative working in co-located project teams is valuable for encouraging teamwork, and how to formalize and streamline stake-holder participation [19].

Creating a learning environment reflecting current best practice for global software engineering projects, is difficult since collaborating teams working at multiple locations are required. Previous research has reviewed global software engineering teaching and the range of learning opportunities created [20] [21]. One approach is to recruit students from multiple universities who then work jointly, forming a virtual team, on a software development project.

C. Geographically Distributed Student Project Configurations

As software development environments are unpredictable, software engineering requires a careful balance between the systematic application of trusted techniques and experimentation with adoption of new practices [22]. Each project must decide on this balance between established and new practices depending upon the specific project, team (size and composition) and level of turbulence.

1) *Project Focus*: Projects involving virtual teams can have differing areas of pedagogical focus. Projects may focus on problem solving and design culminating in production of a report [23]. Multinational teams developing software systems have been used to as part of courses on quality assurance issues [24]. In contrast, software engineering classes have been combined with project work simulating a commercial software development environment [25]. Students working in groups have developed software applications with academic staff acting as clients and providing support and advice with technology, marketing and sales. Groups were selected by staff members and simulated the full software lifecycle.

2) *Project Distribution*: Geographical distribution can involve student teams from multiple universities in the same country [26]. Here each class has different project tasks, responsibilities and expertise. A final software system is produced by students from one of the universities integrating components produced elsewhere. However, geographically distributed projects can provide an opportunity for students to acquire cultural competencies. For example, the initiation phase of a project involving three countries, has focused on using Hofstede et al.'s cultural dimensions [27] to create a framework for the proposed course [28]. Students from four countries involved in international projects have been used to develop a global team learning performance model [29]. The study identifies culture and attitudes, followed by team member experience and grade point average as having the greatest effect on team performance. The authors suggest that the level of interaction is more significant than the amount of time zone overlap [29].

3) *Customer Relationships*: Managing customer relationships is an important aspect of software engineering. Customers on geographically distributed student software projects can be students, lecturers or external clients [30]. Students taking the customer role can face cultural challenges when prioritising and negotiating requirements with student teams [31].

Lecturers acting as customers [32] can manage project scope and ensure student focus for the purposes of assessment. The Global Studio Project involves students from five universities in four countries shadowing a global software development project in Siemens [33]. The Global Studio Project identified the importance of improvisation during the project. A set of real-world project scenarios provided by the Federal Aviation Authority's Simulation and Analysis Group were used for joint projects involving two US universities [34]. Self-forming local teams were established and then paired with a remote team. Team leaders, that remained in post for the duration of the project, acted as the focal point for communication between remote teams. Selection of collaboration tools was left to the discretion of the teams.

4) *Software Development Process Selection*: The software development process can be explicitly mandated in entirety by staff for the project [32], [35] or can be selected by students [36]. In other projects, some more general guidance and advice on software process is given without mandating any particular approach in detail [31].

5) *Software Lifecycle Coverage*: Projects using agile methods tend to cover the entire software lifecycle from requirements, design, implementation through to testing [32]. Such projects require virtual team members to coordinate while producing development artefacts including: requirements specifications, designs, source code, test criteria and test scenarios.

6) *Participant Selection*: Participants on geographically distributed student software projects can be volunteers, selected included automatically as part of a course or module. For some projects staff involvement in participating institutions is primarily focused on "highly-qualified, self motivated" participant selection [37]. Students on the Siemens Global Studio Project were volunteers that had already attended traditional software engineering courses on software processes, requirements, architecture design and human-computer interaction [33].

7) *Personal Contact*: Some projects have funding to support face-to-face contact during an initiation phase of the project [23]. Other projects have obtained limited funding to support face-to-face meetings for staff members, but not sufficient to support each cohort of students [38].

D. Pedagogical Approach

Our approach involves student virtual teams comprising students from two countries [38]. An iterative and incremental approach is used to explore the entire software lifecycle, including requirements analysis, design, implementation, testing and deployment. The project work is supplemented with classroom tuition on software engineering and agile methods. Efforts have been made to concentrate the tuition in the earlier part of the project. A classroom-based action learning approach has been used to evolve the course over three cohorts. We have used open-ended questionnaires to elicit feedback from students and employers.

Open-ended, self-directed learning situations that are founded on enquiry and discovery are known as problem-based learning (PBL) [39]. Problems closely related to a real life scenario are tackled in a group-setting for learners in PBL study schemes [40], helping people learn to solve problems

by applying their knowledge and skills. PBL encourages active experimentation that creates opportunities for students to learn from experience [41].

We view software engineering project work as a form of experiential problem based learning. Project work provides an opportunity to rehearse activities prior to joining a professional community of practice without the pressure of detailed assessment of each phase or sub-task. PBL provides opportunities for the learner to develop skills and construct new understandings in a safe environment. Student learners practice planning skills and are encouraged to show creativity and imagination during the project assignment. Learners are empowered to follow their particular interests within the project, tailoring their learning to build on previous experience. Motivation is improved, when compared with more closely structured learning schemes, because learners control the direction, project scope and detailed implementation of the solution.

III. METHODS

The research questions in this study triggered a mixed method research approach [42]. A classroom-based action research approach has been used to develop the computing curricula module over a period of three years. Action research was supplemented with open-ended surveys that were used to collect data from both students and potential employers of computing graduates.

A. Research Sites

The research was conducted at Robert Gordon University (RGU), UK and the International Institute for IT, Bangalore (IIIT-B), India. RGU achieved university status in 1992, although its origins can be traced to Robert Gordon a merchant and philanthropist who established a hospital for education in 1730. RGU has around 11,000 undergraduate and 6,500 post graduate students. The university is based on a campus in the south western suburbs of the city of Aberdeen, Scotland.

IIIT-B is a graduate school founded in 1999 and awarded deemed university status in 2005. IIIT-B has around 600 postgraduate students and is based in Electronic City, one of India's largest industrial parks, located in the outskirts of Bengaluru, Karnataka. IIIT-B has established close connections with the large and influential Indian and international IT commercial sector.

B. Data Collection

Action research is an iterative methodology for understanding and reflecting upon practice [43] [44]. Action research was used to introduce international project working to our students and effect positive change in practice.

Our action research has comprised three main cycles: extra-curricula pilot [45], credit-bearing pilot [46] and class cohort. The action research cycles we used comprise four phases: (1) problem identification, (2) planning, (3) action and (4) evaluation [44].

Open-ended survey questionnaires were used to elicit feedback from student participants, see Appendix 1. The questionnaire was returned by 10 out of the 12 student participants, in

the credit-bearing pilot and by 19 students in the subsequent iteration.

Global software development activities are completely dependent on the collaborative information technologies that enable productive interaction to take place between the separated parties. Disruption of these technologies almost inevitably leads to breakdown in the channels of communication that teams have set up and a consequent loss of productivity, in some cases, at a critical level. While it is true that issues such as lack of proper team leadership and inadequate management of resources contribute significantly to poor performance in unsuccessful projects, technology plays such a crucial role in enabling collaboration that it provides an obvious starting point for research when considering the educational aspects of GSD. The data collection in the current project therefore focussed on how, and in what way, issues of technology use affected the group interactions, specifically collaboration between the Scottish and Indian contingents.

The initial questions on the questionnaire set out to establish the degree of familiarity of the students with group working in general and their prior use of collaborative technologies to facilitate this type of activity. Data was then sought on how the students went about setting up the collaboration, the main challenges that they faced and the choice of specific technologies they made. Illustrations were given of the type of response that could be given. For example, challenges could be operational (e.g. different time-zones), academic (e.g. different sets of prior knowledge), or social (e.g. different cultural approaches to social interaction). With respect to choice of technology, the main information that was requested was about the factors that led to its adoption, and the decision-making processes that were used to arrive at this choice. The next section attempted to elicit views about specific issues concerning the technology. Some questions asked about the way in which the choice of technology supported the requirements of the task (i.e. technology fit) as well as the effect that the specific choice of technology had on the group interaction (i.e. technology mediation), whether it tended to facilitate some kinds of interaction (e.g. synchronous) but discourage others and how the group assessed its impact on the project. The questionnaire concluded with a question on what the student themselves gained from the experience, as well as asking what the student thought the faculty objectives were in setting up such a course unit, specifically, what employers would think of the activity.

An open-ended standardised online survey was used to collect perspectives of 10 potential employers. Purposeful sampling was used to identify senior executives and business owners. The survey instrument is reproduced in Appendix 2.

IV. STUDENT VIRTUAL TEAM PROJECT

A. Cohort 1 Extra-Curricula Pilot

The main objectives of the extra-curricula pilot were to enable students to:-

- 1) Explore software development in an international team,
- 2) Use social computing (collaboration) software tools to enable communication,

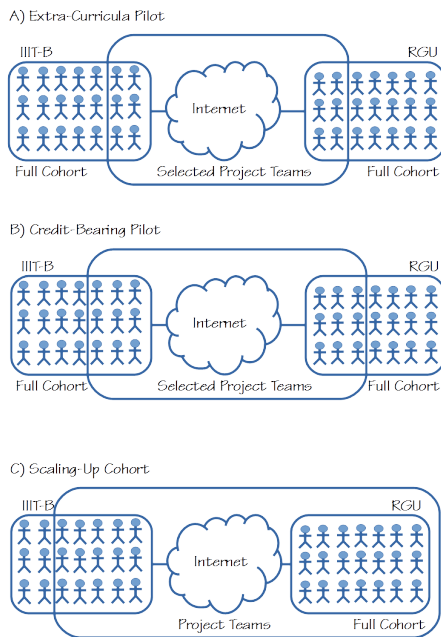


Fig. 1. Team Configurations

- 3) Use an “Agile” software development methodology,
- 4) Assess the impact of culture on communication between team members.

Two Groups of 6 students were established, each group comprising 3 students from each participating institution, as shown in Fig. 1 A. All students in each cohort, including those not engaged on the international group project, conducted their home institution module assessments as usual. The groups comprised an equal number of students from each institution, as shown in Fig. 2 A. The groups selected a project manager which was rotated through the group members during the project. The groups had both co-located and online meetings, with decisions and actions recorded in minutes. Each group produced requirements and design documents, implemented software, testing results, and a project report. Project participants were selected from a set of volunteers, with selection on the basis of previous experience, academic performance, individual class attendance records and communication skills (as assessed by oral interview). Group members undertook agile software development training.

B. Cohort 2 Credit-Bearing Pilot

The credit-bearing pilot participants were again selected from volunteers, however, the international project replaced a co-located group assessed activity, providing a module grade, as shown in Fig. 1 B. Those students that volunteered for the international were assessed on the international project group work. The remaining students in the cohort, i.e. those not engaged on the international group project, conducted the usual module assessments at their home institution. Again the groups comprised an equal number of students from each institution, as shown in Fig. 2 B. Group interactions were scheduled during timetable class time, simplifying the process of arranging real-time conversations between group members. Groups were

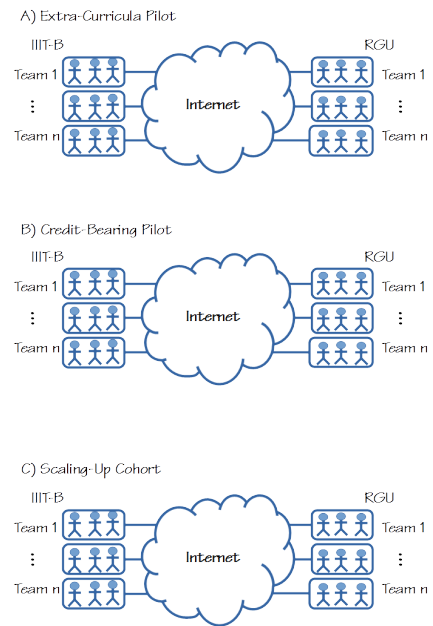


Fig. 2. Team Composition

encouraged to arrange additional meetings outside class. An intensive two day workshop on agile software development methods, based on a commercial short course available from RGU, was provided to group members.

The supervisory team acted as scrum product owners representing the interests of the client and providing a prioritised list of software requirements, while not project managing the team, as such.

The teams built an online survey environment involving mobile phone client software for asking questions which were collated onto a server which stored survey results in a database.

C. Cohort 3 Scaling-up

For the third cohort, the entire UK class was engaged on the project, this was about 20 students from each institution, as shown in Fig. 1 C. The cohort at IIIIT-B is larger, so not all students in the module could be accommodated in the international project activity. The groups comprised an equal number of students from each institution, as shown in Fig. 2 C, although this cohort had a larger number of groups. Training on agile software development methods was provided to group members. This was based on an intensive two-day workshop provided as a commercial short course available from RGU and so was comparable with professional instruction on this type of development methodology. The academic supervisors acted as product owners for the project using the scrum agile method. The product owners provided a prioritized list of the softwares functional requirements. However, the academic supervisors did not project manage the teams, as such.

Each team was required to build a software application. This was achieved in four iterations, as shown in Fig 3. Iteration 0 was for requirements analysis, high level design

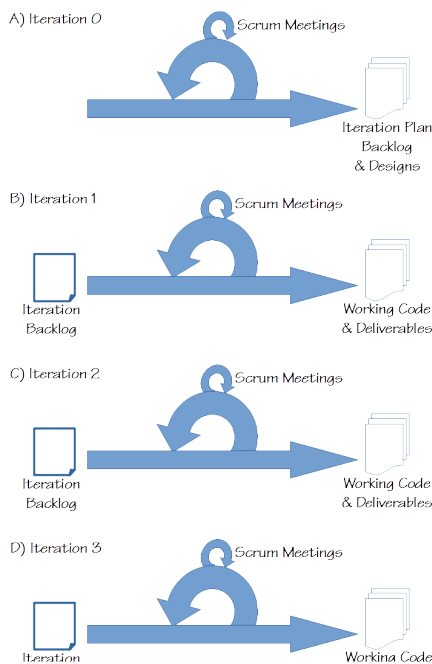


Fig. 3. Sprint Configuration

and iteration planning, see 3 A. The iteration planning involved mapping features planned for development to specific iterations. Iteration planning is usually associated with larger scale agile projects, but here was seen as useful for managing project scope. The remaining three iterations were traditional development sprints using a scrum model, as shown in Fig 3. Scrum meetings, while not conducted daily, usually occurred three or four times a week. The quality of the final software deliverable together with the project management aspects of the activity were assessed and formed the final module grade. The application comprised an online survey environment with a server for storing survey results in a database and mobile phone client software for asking questions.

1) *Student Feedback*: Data collection using the technology use questionnaire described above began with the second iteration of the project and subsequently investigated using a thematic analysis based on the principal conceptual points behind each of the questions.

Even among the small number of participants in the second iteration, the range of experience varied widely with one RGU student claiming no prior experience while two IIT-B students claiming extensive experience of group work. This data was self-reported so the information characterised student self-perceptions rather than describing levels of experience on some objective scale. However, in general, the IIT-B students identified themselves as more experienced group workers than their RGU counterparts. The reported familiarity with collaborative technology was similar between the two cohorts, but again reflecting a wide range of experience within each group. Almost all groups responded that the problem associated with the different time zones was significant but, even so, some response data suggested that individual students within a group did not prioritise time resources issues. Adverse effects on the smooth running of the development groups were seen as a

consequence of the time zone difference but some students did not appear to have an underlying appreciation that such difficulties could be mitigated by greater attention to standard project management tasks. As well as difficulties with time zones, the most widely identified concern was the significant variation in previous programming experience .

Most students in this iteration identified Google groups, Google drive, and Skype as the dominant collaborative technologies used, which was surprising given that these technologies were not specifically designed for this type of project work. This was, however, consonant with a general desire to prioritise robustness and ease of use over technical sophistication when adopting specific technologies. In general, students stated that the decision-making process around technology adoption was one of seeking consensus, although some students reported that the views of perceived experts were given extra weight within the consensus-seeking procedure. No student indicated how conflict resolution was achieved if this process broke down irretrievably. There were many responses concerning technology that was initially adopted but then discarded due to installation difficulties or lack of available time to engage with it on a professional level. One example of this was the web-based hosting service, GitHub which was acknowledged to be potentially useful but was found to be difficult to set up and was therefore discarded, first in favour of Subversion, and then, when this also proved too difficult, in favour of a low-tech solution based on Google Groups. Nevertheless, the students themselves noted that decisions about the lack of adoption of software were sometimes a source of problems at a later stage.

When asked about the benefits of their experience of GSD, both the UK and Indian cohorts of students tended to view the experience of developing the software deliverable within a distributed team as the primary learning objective. When asked to conjecture the opinion of other stakeholders, the students focussed exclusively on employers but the views expressed were positive. There was a general appreciation that experience of working in such distributed teams and of using a range of collaborative technology would make the student more attractive to potential employers and would contribute positively to their CV. This latter point is slightly at odds with the experience of many of the students where much of the industry-specific collaborative and software development technology initially adopted was later dispensed with in favour of more ad-hoc solutions to the problem of collaborative software engineering.

In general, however, there was enthusiasm from students about the concept of the international group project group-work with an Indian university. It was an interesting and insightful experience (Student1), “an exciting module with the aspect of working with a group from another country” (Student2) and “a very interesting course” (Student3). However, there were concerns about the logistics and implementation of the module “there wasn’t much teaching time” (Student4), “no guidance in place for the technologies we were expected to use” (Student4), “a time consuming module” (Student5) and “the collaboration had too many loopholes (meeting times, course schedule, etc.)” (Student6).

In the third yearly iteration of the course unit, data was again collected using the technology-use questionnaires. This

time, responses were received from six RGU and thirteen IIT-B students. The submissions of the former were, in general, somewhat more substantial than the latter although less numerous. Both RGU and IIT-B students gave details of a range of experience of working in teams. At RGU, undergraduate participants from the class reported significant experience of technical and project management activity whereas when MSc students reported team-working exercises, they did so mainly in the context of non-Computing groups. The situation at IIT-B was similar with students reporting a range of activities with a group work component.

When asked to comment on their prior experience of collaborative technology, both cohorts focussed on examples of general communications software (e.g. Skype or Google Hangouts), file sharing facilities (such as Dropbox or Google Drive) and social media applications. While some students had heard of more specialised software such as Github, use of this as collaborative tools was restricted to a very few students.

As expected from the returns from the questionnaire in the previous iteration, the single biggest challenge to collaborative working was reported to be the difference in time zones, which was stated to have a significant impact on the majority of project management issues. Both groups of students also expressed frustration at the detrimental effects of lack of appropriate hardware and network connectivity problems, which resulted in disagreement over the use of technology.

As in previous cohorts, the lack of a baseline level of prior academic knowledge and technical expertise was also cited as an important negative factor. The problems associated with the wide variation in individuals' prior skill set appear to underlie, or at least have contributed to, communication difficulties reported by students in both locations. This was sometimes manifested as a perception by some students of a lack of seriousness on the part of individuals in the other team. Both sets of students identified the need for clearly defined aims, for the software development exercise, and also for the learning objectives of the course unit in which the activity took place. The lack of a common set of course-unit learning objectives meant that agreement on a development methodology was more difficult, which again, led to frustration in some groups.

So far as technologies used for the task, the groups found that those used for local communication and development appeared to work well but there were, perhaps inevitably, less positive reactions to synchronous communications which tried to mediate between the distributed groups. All teams decided to use general communication tools such as Skype or Google Hangouts, and while these were effective, there were problems with the robustness of the network infrastructure in both universities. Both groups reported limitations of web or cloud-based sharing facilities and many groups appear to have resorted to using conventional asynchronous technologies such as email to share code. There were also some reports of the use of social media software to accomplish project management tasks.

Unlike the previous cohort, the responses revealed that a variety of decision-making processes were employed to agree on specific use of technology, ranging from systems of majority voting to decisions made by a single member

TABLE I. EMPLOYER ATTITUDES TO STUDENT INTERNATIONAL TEAM WORKING

Employer Attitudes to International Team Working	Agree or strongly agree	Neutral	Disagree or strongly disagree
Graduates should learn some facts and knowledge	10	0	0
Graduates should learn through first hand experience	9	1	0
Graduates should be assessed	5	4	1

perceived to be the group "expert". Nevertheless, it appears that consensus was again valued as a procedural goal.

There was surprisingly little analysis or comment by users in any of the groups about how the use of technology impacted on the types of interaction between group members within a co-located group or with the separated part of the team. Technology was assessed from a purely functional perspective and fitness for purpose was assessed almost solely on something being operational at the appropriate time. If a technology worked, then it was perceived to be good regardless of the effects on collaboration. There was also little analysis of alternatives: if something worked initially then the technology was used without subsequent investigation of whether it was the best tool for the activity in question.

While both Indian and Scottish groups of students continued to play down any explicit reference to culture, the third iteration did start to show some differences in perception between the two groups. There was a difference in how groups tended to report miscommunication how this was attributed. We discuss this in the next section.

Despite the difficulties encountered in this cohort, both sets of students reported significant benefits from participation in the project, although, in almost all cases, increases in technical competences was cited as the reason for this learning gain. Increased technical knowledge of Java/Android programming and web service development were most often cited as the important learning objectives attained by both groups of students. Exposure to Agile methods was reported as another positive experience with some development of practical skills in project management, such as team leadership, resource allocation, conflict management and time keeping, being seen as relevant. In addition, both groups of students stated that they thought that employers would look very favourably on this type of activity and that it would significantly enhance their employability compared to their peers who had not undertaken the exercise. When asked about the reasons that such a course unit was developed, one student stated that: "I believe the aim of the staff when they set up the collaboration was to simulate the real world environment and prepare us as a modern software developer capable of team-work, capable of peer-learning, capable of handling difficult situations and characters, capable of researching and sharing skills knowledge and understanding" (Student7).

2) *Employer Survey Findings:* Employers were supportive of including international team working in the curriculum. Table I shows that employers were more positive about learning facts and knowledge and participating in project work, than they were about assessment of those skills.

Table II shows that most employers thought that interna-

TABLE II. EMPLOYER ATTITUDES TO EMPLOYABILITY

Employer Attitudes to Employability	More or significantly more employable	Neutral	Less or significantly less employable
Would graduates with these skills be more, or less, employable?	8	2	0

tional project work would enhance skills leading to improved employability.

The key international team work issues graduates should be aware of are:

- Developing rapport with remote work colleagues (9 respondents)
- Delays caused by the communication overheads of cross site working (7 respondents)
- Cultural awareness when dealing with others (7 respondents)
- Impact of work item coupling between sites (5 respondents)

A minority of respondents thought software tools, either to support collaborative working or communications, were issues for graduates to be aware of. Other issues respondents identified included:

- Understanding of time zones
- Problem solving...balance people, process, technology, and x-cultural issues
- Need for correct, comprehensive, clear and easily readable written communication
- Value systems, influence of personality differences, preconceptions/stereotyping and working styles, and
- Language and semantics in multi-lingual environments.

One respondent was sceptical that virtual team working would make much difference at undergraduate level, feeling “this would come from on-the-job experience” Other respondents argued that “both the ‘soft’ and the ‘hard’ aspects are required to implement teamwork effectively”, “building skills in cultural awareness” and “exposing students to different realities is simply priceless” and “will help employees to get ahead faster.” However, these skills would not compensate for “any weak points on the ‘hard’ technical skills side.”

V. DISCUSSION AND RECOMMENDATIONS

Two universities have collaborated to provide a joint software engineering project using virtual student teams. The first cohort comprised volunteers working on an extra-curricula basis. The second comprised volunteers with the virtual team activity substituted for a co-located coursework assignment at both universities. The third cohort comprised the full class of UK students match with an equivalent number of team members from India.

We identify several lessons from our experience over the three years:

- pilot projects help win buy-in from institutional stakeholders and build supervisory staff expertise in supporting student virtual teams,
- contrasting assessment demands between institutions must be reconciled,
- detailed joint timetable planning helps to reconcile different institution’s teaching calendars,
- provide a project management framework to encourage learning about development processes,
- support phased delivery to minimise project risk and
- carefully manage project scope to ensure deliverables are achievable.

Some students seemed to have difficulty reducing the scope of functionality they had committed to produce, as the project unfolded. Unforeseen problems and challenges hampered progress towards the goals they had set themselves, creating stress for some team members. This stress was subsequently alleviated when the team members reduced project scope in later iterations.

One aspect of the responses to the questionnaire that is perhaps worth mentioning is the relatively low levels of reflective activity that students reported concerning the impact of technology on the project activities. In general, collaborative technology was seen in purely functional terms as a means to accomplish a specific, technical, end. There was almost no consideration of wider aspects of technology use such as its role in providing a mechanism for group coherence, both among the co-located team members, and between the geographically separated group. The effects of technology were seen in instrumental terms as working to produce the final software artefact and there was little thought of the reverse effects that such use could have in shaping group perceptions of the project task.

In the second cohort, this relatively low level of analysis about technology extended to the sphere of inter-cultural competence. In general, apart from some complaints between the groups about differing views of the project deliverable and the method to achieve them, there were not many comments that addressed differences in work practices due to cultural variation between India and Scotland. While there were some comments that expressed a degree of exasperation about physical and temporal differences (time-zones, differences in academic calendar, differences in the stated learning objectives between the Scottish module and the Indian one), there were no comments which concerned the cultural mores of the two teams.

It is interesting to note that in the third cohort, the students from both universities appeared to downplay cultural differences between the UK and Indian groups, where common points of disagreement occurred, the specific nature of these was reported differently depending on whether the student was from the UK or India. One example of this would be problems associated with people being given tasks that they were not able to complete. UK students appeared to interpret this in

terms of lack of honesty whereas Indian students interpreted it as lack of commitment. For example, it was reported that the systems that provided synchronous video communication, such as Skype, placed some impediments to frank and honest discussion. Some UK students stated that such communication appeared to put some of their Indian colleagues, who may not have had sufficient expertise in some particular aspect of the project, in a difficult situation and rather than admit this or state that they did not have the appropriate skillset to accomplish specific tasks, they would agree to do things that they could not complete. Furthermore, misinterpretation of voice communication was cited as one reason for delay as often this would result in requests to resend queries at a later date in the textual form. In general, while the Indian students did not report these observations, there were corresponding complaints from some individuals in the IIIT-B teams that RGU students who had failed to perform certain tasks, which had been agreed at the team meetings, showing a lack of sufficient commitment to the project. In both cases, such behaviour appears to have led to an elevated degree of distrust in some groups.

It is, of course, difficult to draw hard conclusions from small sample sizes but it may indicate a difference in the perceived roots of failure in the two classes and may be something that could be mitigated by an increased attention to the development of inter-cultural competence.

The fourth cohort of virtual project teams is running at the time of writing. We are keen to gain a deeper understanding of the specific student knowledge, skills and attitudes affected by the project. We are using a skills inventory derived from a well-established model of industry software development roles.

VI. CONCLUSIONS

Over the last three years, Robert Gordon University, UK and the International Institute for IT Bangalore, India have collaborated to develop a joint software engineering project involving remote working in virtual student teams. The students work together in virtual teams and never met face-to-face.

In this paper we have provided an account of introducing global software engineering into the computing curriculum. We introduced the activity initially as an extra-curricula activity using volunteers. For the second cohort, the virtual team working project used volunteers but replaced another internal assessed activity. The third cohort comprised the entire UK class of about 20 students. This progressive approach to introducing international group work, minimised risk and helped staff members acquire virtual project management skills in the context of the classes. Minimal additional resources and support was available to add the international component to the modules, although travel funds enabled a staff member from each institution to meet and establish the joint project.

We found students from all three cohorts supportive of the concept of the virtual team project. Although each cohort seemed to face new logistical challenges when collaborating with students from another institution. Reconciling semester dates, scheduled class times and assessment objectives have all been challenging at various times. As always with student project work, there is a tension between teacher-led tuition and student-led self-study. Some students welcome tuition in which instructors provide facts and knowledge. While, many

of the skills required for project work come from self-study and self-organising teams, learning from doing.

We found employers were supportive of global software engineering in university curricula, suggesting knowledge and skills would strengthen student employability prospects. Employers argued that students should develop skills in developing rapport with remote work colleagues, managing delays caused by the communication overheads of cross site working and cultural awareness when dealing with others.

In addition to the progressive approach to introducing international group work, we have identified four key principles for global software engineering student projects: reconcile contrasting assessment demands between institutions, create a detailed joint timetable to reconcile teaching calendars, provide a project management framework to support phased delivery and carefully manage project scope.

ACKNOWLEDGMENT

This work is part of an ongoing project funded by the UK Higher Education Authority. The authors would like to thank the HEA for its continued support. The research also benefited in part from travel funding to JMB from the UK Deputy High Commission, Bangalore, Science and Innovation Network and the British Council UKIERI scheme. We are grateful to all the students that have participated in the international projects. Their feedback in various forms have been insightful for the course teams at both institutions.

REFERENCES

- [1] Joseph Stiglitz, *Globalization and Its Discontents*. Penguin, 2003.
- [2] J. D. Herbsleb and A. Mockus, "An Empirical Study of Speed and Communication in Globally Distributed Software Development," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 481–494, 2003.
- [3] M. Y. Vardi, "Globalization and Offshoring of Software revisited," *Commun. ACM*, vol. 53, no. 5, pp. 5–5, May 2010. [Online]. Available: <http://doi.acm.org/10.1145/1735223.1735224>
- [4] J. Herbsleb and D. Moitra, "Global software development," *Software, IEEE*, vol. 18, no. 2, pp. 16–20, Apr. 2001.
- [5] C. Ebert, *Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing*, 1st ed. Hoboken, N.J: Wiley-Blackwell, Dec. 2011.
- [6] J. Stapleton, *DSDM: Dynamic Systems Development Method*. Harlow, England: Addison Wesley, 1997.
- [7] P. Coad, E. LeFebvre, and J. D. Luca, *Java Modeling in Color*. Englewood Cliffs, NJ: Prentice Hall, 1999.
- [8] A. Cockburn, *Agile Software Development*. Reading, MA: Addison Wesley, 2001.
- [9] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [10] K. Beck and C. Andres, *Extreme Programming Explained*, 2nd ed. Addison Wesley, Nov. 2004.
- [11] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.
- [12] J. M. Bass, "Agile method tailoring in distributed enterprises: Product owner teams," in *Proc. IEEE 8th Int. Conf. on Global Software Engineering*, Bari, Italy, Aug. 2013, pp. 154–63.
- [13] —, "Scrum Master Activities: Process Tailoring in Large Enterprise Projects," in *Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on*. Shanghai, China: IEEE, Aug. 2014, pp. 6–15.

- [14] S. de Cesare, M. Lycett, R. D. Macredie, C. Patel, and R. Paul, "Examining perceptions of agility in software development practice," *Commun. ACM*, vol. 53, no. 6, pp. 126–130, Jun. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1743546.1743580>
- [15] T. Dyba and T. Dingsoyr, "What do we know about agile software development?" *IEEE Software*, vol. 26, no. 5, pp. 6–9, 2009.
- [16] N. R. Mead, "Software engineering education: How far we've come and how far we have to go," *Journal of Systems and Software*, vol. 82, no. 4, pp. 571 – 575, 2009, special Issue: Selected papers from the 2008 {IEEE} Conference on Software Engineering Education and Training (CSEET08). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121208002756>
- [17] ACM/IEEE-CS Joint Task Force on Computing Curricula, "Computer Science Curricula 2013," ACM Press and IEEE Computer Society Press, Tech. Rep., Dec. 2013. [Online]. Available: <http://dx.doi.org/10.1145/2534860>
- [18] J. Bass and R. Heeks, "Changing computing curricula in African universities: Evaluating progress and challenges via design-reality gap analysis," *The Electronic Journal of Information Systems in Developing Countries*, vol. 48, no. 5, pp. 1–39, 2011.
- [19] C.-Y. Chen and P. P. Chong, "Software engineering education: A study on conducting collaborative senior project development," *Journal of Systems and Software*, vol. 84, no. 3, pp. 479 – 491, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121210002931>
- [20] M. J. Monasor, A. Vizcaino, M. Piattini, and I. Caballero, "Preparing Students and Engineers for Global Software Development: A Systematic Review," *2012 IEEE Seventh International Conference on Global Software Engineering*, vol. 0, pp. 177–186, 2010.
- [21] L. L. Fortaleza, T. Conte, S. Marczak, and R. Prikladnicki, "Towards a gse international teaching network: Mapping global software engineering courses," in *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), 2012*, June 2012, pp. 1–5.
- [22] T. Dyba, "Improvisation in Small Software Organizations," *IEEE Software*, vol. 17, no. 5, pp. 82–87, 2000.
- [23] M. Daniels, A. Cajander, A. Pears, and T. Clear, "Engineering education research in practice: Evolving use of open ended group projects as a pedagogical strategy for developing skills in global collaboration," *International Journal of Engineering Education*, vol. 26, pp. 795–806, 2010.
- [24] O. Gotel, V. Kulkarni, C. Scharff, and L. Neak, "Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development," in *Software Engineering Education and Training, 2008. CSEET '08. IEEE 21st Conference on*, Apr. 2008, pp. 33–40.
- [25] D. Petkovic, G. Thompson, and R. Todtenhoefer, "Teaching Practical Software Engineering and Global Software Engineering: Evaluation and Comparison," in *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ser. ITICSE '06. New York, NY, USA: ACM, 2006, pp. 294–298. [Online]. Available: <http://doi.acm.org/10.1145/1140124.1140202>
- [26] L. J. Burnell, J. W. Priest, and J. B. Durrett, "Teaching distributed multidisciplinary software development," *Software, IEEE*, vol. 19, no. 5, pp. 86–93, Sep. 2002.
- [27] G. Hofstede, G. J. Hofstede, and M. Minkov, *Cultures and Organizations: Software of the Mind, Third Edition*, 3rd ed. New York: McGraw-Hill Professional, Jun. 2010.
- [28] A. Zeid and R. El-Bahey, "Establishing a global software development course: A cultural perspective," in *Frontiers in Education Conference, 2013 IEEE*, Oct. 2013, pp. 1695–1701.
- [29] K. Swigger, F. Nur Aplaslan, V. Lopez, R. Brazile, G. Dafoulas, and F. C. Serce, "Structural Factors That Affect Global Software Development Learning Team Performance," in *Proceedings of the Special Interest Group on Management Information System's 47th Annual Conference on Computer Personnel Research*, ser. SIGMIS CPR '09. New York, NY, USA: ACM, 2009, pp. 187–196. [Online]. Available: <http://doi.acm.org/10.1145/1542130.1542167>
- [30] I. Bosnić, I. Čavrak, M. Žagar, R. Land, and I. Crnković, "Customers' Role in Teaching Distributed Software Development," in *Software Engineering Education and Training (CSEET), 2010 23rd IEEE Conference on*, Mar. 2010, pp. 73–80.
- [31] O. Gotel, V. Kulkarni, L. Neak, C. Scharff, and S. Seng, "Introducing Global Supply Chains into Software Engineering Education," in *Software Engineering Approaches for Offshore and Outsourced Development*, ser. Lecture Notes in Computer Science, B. Meyer and M. Joseph, Eds. Springer Berlin Heidelberg, 2007, vol. 4716, pp. 44–58. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75542-5_4
- [32] M. Paasivaara, C. Lassenius, D. Damian, P. Rätty, and A. Schröter, "Teaching students global software engineering skills using distributed scrum," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 1128–1137. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2486788.2486945>
- [33] I. Richardson, A. E. Milewski, N. Mullick, and P. Keil, "Distributed Development: An Education Perspective on the Global Studio Project," in *Proceedings of the 28th International Conference on Software Engineering*, ser. ICSE '06. New York, NY, USA: ACM, 2006, pp. 679–684. [Online]. Available: <http://doi.acm.org/10.1145/1134285.1134390>
- [34] A. Rusu, A. Rusu, R. Docimo, C. Santiago, and M. Paglione, "Academia-academia-industry Collaborations on Software Engineering Projects Using Local-remote Teams," in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '09. New York, NY, USA: ACM, 2009, pp. 301–305. [Online]. Available: <http://doi.acm.org/10.1145/1508865.1508975>
- [35] N. Mullick, M. Bass, Z. El Houada, D. J. Paulish, M. Cataldo, J. Herb-sleb, and L. Bass, "Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure," in *Global Software Engineering, 2006. ICGSE '06. International Conference on*, Oct. 2006, pp. 203–212.
- [36] I. Bosnić, I. Čavrak, M. Orlić, M. Žagar, and I. Crnković, "Avoiding scylla and charybdis in distributed software development course," in *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*, ser. CTGDSD '11. New York, NY, USA: ACM, 2011, pp. 26–30. [Online]. Available: <http://doi.acm.org/10.1145/1984665.1984671>
- [37] R. Holmes, M. Craig, K. Reid, and E. Stroulia, "Lessons learned managing distributed software engineering courses," in *Companion Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE Companion 2014. New York, NY, USA: ACM, 2014, pp. 321–324. [Online]. Available: <http://doi.acm.org/10.1145/2591062.2591160>
- [38] R. McDermott, J. M. Bass, and J. T. Lalchandani, "The Learner Experience of Student-Led International Group Project Work in Software Engineering," in *44th Annual Frontiers in Education Conference*. Madrid, Spain: IEEE, Oct. 2014, pp. 1181–1188.
- [39] D. Boud and G. Felelitti, *The Challenge of Problem-Based Learning*, 2nd ed. London: Kogan Page Ltd., 1997.
- [40] L. Thorley and R. Gregory, *Using Group-Based Learning in Higher Education*. London: Kogan Page Ltd., 1994.
- [41] David A. Kolb, *Experiential Learning: Experience as a Source of Learning*. Englewood Cliffs, NJ: Prentice Hall, 1984.
- [42] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 4th ed. Thousand Oaks: SAGE Publications, Inc, May 2013.
- [43] K. Lewin, *Action Research and Minority Problems*, 2nd ed., ser. Course Team (Eds), The Action Research Reader, Action Research in the Curriculum. Victoria, Australia: Deakin University Press, 1982, journal of Social Issues, Vol 2, 1946.
- [44] J. McNiff, *Action Research: Principles and Practice*. London, UK: Macmillan Education, 1988.
- [45] J. M. Bass, C. Ramanathan, and J. T. Lalchandani, "Software engineering: learning, employment and globalization." Birmingham, UK: Higher Education Academy, 2013. [Online]. Available: <http://www.heacademy.ac.uk>
- [46] J. M. Bass, J. T. Lalchandani, and R. McDermott, "International group work in software engineering." Higher Education Academy, Apr. 2014. [Online]. Available: <http://www.heacademy.ac.uk/assets/documents/stem-conference/Conference-proceedings-2014/Computing/COMP-204-P.pdf>

APPENDIX 1 OPEN-ENDED STUDENT QUESTIONNAIRE

The first two questions look at your prior experience of working in groups and using collaborative technology.

- 1) On a scale of 1 to 5, please rate the degree to which you considered yourself to have been prepared for the group work aspects of this particular module.
Please describe in a short paragraph any previous experience of group work.
- 2) On a scale of 1 to 5, please rate your degree of familiarity with Collaborative Technology (CT), i.e. any hardware or software technology that enables you to work together across a distance.
Please describe in a short paragraph your previous experience with CT. The next two questions relate to how your group went about setting up the collaborative project.
- 3) What were the initial challenges you found in working in this collaboration?
These challenges may have been operational (e.g. different time-zones,), academic (e.g. different sets of prior knowledge,), social (e.g. different cultural approaches to social interaction,) or something else.
- 4) Which collaborative technology (or set of technologies) did you use in the project? How did the group make the decision about which technology to use? What were the main factors that led to its adoption? Were there any negative aspects associated with the decision? Did you use any structured method to analyse or prioritise the various factors? Was the decision made by consensus, by majority, ?
These two questions relate to how your group used the technology as the project progressed.
- 5) How good was the match between the requirements of your task and the capabilities of the technology you chose to support it?
This is known as Technology Fit. There are three areas on which you might comment: how well did it provide communication support; how did it support the structuring of the project management processes; and how did it support the development (design, coding, ...) of the software objects that form the basis of the project deliverable?
- 6) How did the technology affect the way you interacted with other members of the group? Please give examples.
Again, there are three areas on which you might comment: what effect (if any) did the capability to communicate with other group members have on individuals/the group; what were the effects of support for the structuring of the project processes; and what were the effects on the group of having support for developing the project deliverables?
These questions ask you to reflect on your use of technology and on any benefits you think the project has brought to you:
- 7) How did you assess the manner in which the technology affected your interaction with other members

of the group?

Did the technology tend to facilitate some kinds of interaction (e.g. synchronous) but discourage others? How did you assess its impact on the way you could carry out the project processes?

- 8) What benefits do you think you have gained from this exercise? Are they primarily academic (application of new technical skills, learning new subject content,) or professional (development of team-working, project management skills,)?
The final question looks at perceptions of project goals by different stakeholders. This is an important question so please give as full an answer as possible.
- 9) What do you think were the aims of this module? What were your personal objectives in taking it? Has the module been successful from your own point of view? What do you think the aims of staff were when they set up the collaboration? Do think that they feel it has been successful? How do you think employers would view this kind of activity? Is there anything that could be done to improve it?

APPENDIX 2 OPEN-ENDED EMPLOYER QUESTIONNAIRE

- 1) Where is your usual place of work?
[Onshore; Offshore]
- 2) I would like graduates to be aware of the following international team work issues. Please choose any that apply.
 - Delays caused by the communication overheads of cross site working
 - Impact of work item coupling between sites
 - Software tools to support collaborative working (e.g. Wikis, shared repositories, etc)
 - Software tools to support video and audio communication
 - Cultural awareness when dealing with others
 - Developing rapport with remote work colleagues
 - Other (please specify)
- 3) I would like graduates to have learned some facts and knowledge about international team working [Strongly Agree, Agree, Neither Agree Nor Disagree, Disagree, Strongly Disagree]
- 4) I would like graduates to have learned about international team working through first hand experience [Strongly Agree, Agree, Neither Agree Nor Disagree, Disagree, Strongly Disagree]
- 5) I would like international team working to form part of graduate assessment [Strongly Agree, Agree, Neither Agree Nor Disagree, Disagree, Strongly Disagree]
- 6) Would graduates with these skills be more, or less, employable? [Significantly More Employable, More Employable, Neither More Nor Less Employable, Less Employable, Significantly Less Employable]
- 7) Any other comments about international team working?