Introducing Global Supply Chains into Software Engineering Education

Olly Gotel[1], Vidya Kulkarni[2], Long Chrea Neak[3], Christelle Scharff[1], and Sopheap Seng[3]

[1] Pace University, Seidenberg School of Computer Science and Information Systems, Department of Computer Science, New York, NY, USA
{ogotel,cscharff}@pace.edu
[2] University of Delhi, Computer Science Department, New Delhi, India
vkulkarni@cs.du.ac.in
[3] Institute of Technology of Cambodia, Computer Science Department, Phnom Penh, Cambodia
{longchrea.neak,sopheap.seng}@itc.edu.kh

Abstract. This paper describes lessons from running software development projects across three globally distributed educational institutions. What was innovative about this study was that two of the institutions were located in service providing countries, conventional onshore/offshore roles were reversed, and students were exposed to the realities of global supply chain management. Three teams of US students were tasked to develop three different software products for Cambodian clients, while sub-contracting the database component to third-party teams of Indian students. This paper details the role of the three institutions, the prerequisites for planning and logistics for running such educational projects, and summarises the findings, while drawing broader parallels with the commercial world of offshore and outsourced development. It ends with recommendations for software engineering education to better reflect the needs and skills demanded of right sourcing in the global marketplace. These extend more generally to global software engineering.

Keywords: Global Software Development, Software Engineering Education, Supply Chain.

1 Introduction and Background

With the need to reduce development cost and improve quality, software products are increasingly developed via collaborations across people, organizations and countries. The challenges facing such globally distributed software development projects have been studied and reported in the literature, generally focusing on the economic, technical, organizational and cultural issues [4,15,16]. The September 2006 issue of IEEE Computer: *Global Software Development: How Far Have We Come?* explicitly captures the state of the practice in global software development.

B. Meyer and M. Joseph (Eds.): SEAFOOD 2007, LNCS 4716, pp. 44–58, 2007.

A global provisioning model often implies the existence of a prime (or lead) contractor and a chain of sub-contractors. Increasingly, it is likely that software houses will outsource well-defined components of their contracts to smaller companies, and even to cheaper service providing countries. The motivation is not always about sourcing software development in the cheapest place, but about sourcing it where there are the requisite skills and continuous coverage [19].

This model has an obvious implication for computer science and software engineering education around the world [1,2,9,10,21,22]. In these disciplines, students need to be exposed to the realities if they are to develop the differentiating 'softer' skills they will need [14]. The supply chain model of software development requires students to learn how to divide up a project into component parts for different parties to work on, these parties possibly being distributed across time zones and cultures. This obviously relies on a shared and systematic process. In addition, it requires students to learn about scoping and delineating boundaries, eliciting requirements from remote clients, communicating an understanding of requirements back to clients and then on to third-parties (not the same skill sets), and learning about the testing and integration needed to assemble a working whole product when the component aspects are not under their total control. A number of courses are now beginning to reflect some of these realities, focusing on requirements engineering [3,6], the development of software in a global context [17,18] and the provision of global projects from within corporations [20]. However, few pioneering courses appear to involve service providing countries, such as India and China and, to our knowledge, do not simulate supply chain development.

Since 2005, the focus of the US and Cambodian partners' undergraduate capstone software engineering courses has been global software development [11]. In 2005, teams of Pace University and Institute of Technology of Cambodia (ITC) students worked together to develop software products for the Cambodian market. The student projects were organized so that: (a) the Cambodian students acted as clients and end-users - they knew the problem the proposed system was to tackle, the environment it was to operate in and had the authority to accept the work of the providers (or not); and (b) the Pace University students acted as providers - it was their responsibility to 'capture' the requirements for the system, propose design options, develop the selected design and test the eventual system, while also handling requirements changes. In 2006, this model[1] was extended to include students from the University of Delhi. These students had expertise in database design, so the concept of a prime contractor and third-party supplier (sub-contractor) was introduced to reflect these skills. While the Cambodian students remained as clients, the provisioning of the solution was changed. The Pace University students sub-contracted part of the system design and development to the students from India, while also managing the end-to-end contract. This paper describes our 2006 study which examined the following questions:

[1] The web page dedicated to this study can be found at *http://www.csis.pace.edu/∼scharff/cs3892006*

a) What is required to set up and run a global distributed project of this chained nature in an educational context?
b) How does the use of a supply chain impact the requirements engineering process, notably the handling of change and the assumptions made by the various parties?
c) What are the perceived communication and coordination issues along the supply chain, and do they differ?
d) Is there the perception of a 'global' team across a distributed supply chain and, if so, what social activities can foster this unity?
e) What are the main differences in interaction between prime contractors and sub-contractors and between prime contractors and clients?
f) How does the nature of this interaction impact project quality? 'Quality' here refers to the conformance of the end product to the clients' specified and perceived requirements [5], ascertained by final acceptance or rejection.

The remainder of the paper is organized as follows. Section 2 establishes the context of the study and Section 3 describes the preparation that is required to set up multi-institution collaborations of this nature (addressing question a above). The key findings from the study, with respect to questions b-f, are summarized and discussed in Section 4. Section 5 provides conclusions and recommendations to assist other institutions in reflecting the realities of the global marketplace in the curriculum.

2 Context

In this section, we describe the three institutions this study is based on, their roles in the global supply chain, the courses targeted by the study, the software that were developed, the composition of the global teams, and the process and technology that was used to enable collaboration in this global setting.

2.1 The Institutions

The Institute of Technology of Cambodia (ITC) (*http://www.itc.edu.kh*) is a leading semi-public higher education school in Phnom Penh accepting students on a competitive exam. The educational system is organized by trimesters. Cohorts of five-year engineering students go through the programs with about thirty hours of core science classes a week during the first two years, then specialize and attend about thirty hours of classes a week in their speciality.

Pace University (*http://www.pace.edu*) is a private university strategically located in New York offering programs at the undergraduate and graduate level. The educational system is organized by two main semesters. Undergraduate students go through the programs with a load of about nine to fifteen hours of computer science and liberal arts classes a week. The concept of a yearly cohort is less prevalent since students can take courses at their own pace, though most complete within four years.

The University of Delhi (*http://www.du.ac.in/*) is one of the prestigious public institutions in India granting Bachelors, Masters and Ph.D. degrees. Admission to all programs is through entrance examination and limited to thirty students in each program. Like the ITC system, students enrolled in these programs attend classes for about thirty hours a week and the educational system is organized by trimesters.

2.2 Institutional Roles and Targeted Courses

The study described in this paper involved a capstone undergraduate 'Software Engineering' course taken by junior (third year) and senior (fourth year) Pace University computer science students, a 'Software Engineering' course for fourth year ITC computer science undergraduate students and a 'Database System Implementation' course for first year M.Sc. in computer science students at the University of Delhi. The intention behind this effort was to provide students with a realistic co-production experience where software products have to be engineered by global partners with disparate skills and expertise. Note that the study ran from January through May 2006, so was aligned with the US spring semester.

ITC Students are Clients, Testers and Translators. Their responsibilities were to describe the software they wanted to be built and the context in which it was to operate. They also had to review and give feedback on the requirements, design and testing documents, test the software and submit bug reports, and ultimately deliver the software in French and Khmer for a Cambodian audience[2]. Their responsibilities also included reporting on the problems arising from working with the US students. At the end of the semester, the Cambodian students had to assess the software developed by the US students (with Indian sub-contracting) and to compare this with the software developed solely by the Indian students (see Indian roles later in this section).

US Students are Developers and Lead Contactors. Their responsibilities were to capture the requirements from the clients and produce an agreed specification, propose design options that subcontract part of the system design and development to Indian students, implement the software and test it, while concurrently handling requirements changes, integrating feedback and managing the end-to-end contract. At the end of the semester, the US students had to deliver the software to their clients. Additionally, the US students were required to maintain a web page for the project, report on the problems arising when working with the Cambodian and Indian students, answer a weekly questionnaire concerning communications and requirements changes, archive all emails and chat sessions, and document their experience with the software engineering process and communication protocols they followed.

[2] Due to time constraints the software was only delivered in English.

Indian Students are Third-party Suppliers. Their responsibilities were to provide the US students with a database design and the corresponding SQL code to be integrated in the overall system design. Their responsibilities also included reporting on the problems arising from working with the US students[3].

Global Supply Chain Scenario. This study was set up such that there were no contacts between India and Cambodia, as initially the Indian students were intended to act as pure suppliers in the project and joined the project one month after the Cambodian students. The Indian students had database expertise that was lacking in some of the US students' repertoire. The study was also explicitly designed to promote a reversal of conventional onshore/offshore roles for a number of reasons. Firstly, to give the US students the opportunity to find out what it would be like to be on the development side of the onshore/offshore scenario, and because the US students do not always have enough programming opportunities that demand version control and large scale integration. Secondly, to give Cambodian students exposure to the empowering position of being a customer and hopefully cultivating entrepreneurship. Lastly, to give Indian students a realistic experience of working with pre-specified requirements and learning how to work smoothly as part of a chain. All the students were to experience working with others from a different culture.

2.3 Teams

In the spring of 2006, the class at Pace University comprised eleven students, the class at ITC comprised sixteen students, and the class at the University of Delhi comprised thirty students. Of the Indian students, only six were part of the study - the six students who had obtained the highest score on the first midterm exam of the class. The study thereby consisted of three global teams each composed of ten to twelve students distributed amongst the three locations: three to four students from the US acting as prime contractors; five to six Cambodian students acting as clients; and two Indian students acting as sub-contractors. The term local team was used to refer to co-located team members. The students were free to choose their own local teams. The US and Indian students chose their global team partners in Cambodia and the US respectively based on project preferences. Students were assigned roles in the teams (e.g. team, communication and quality assurance leaders).

2.4 Projects

Three Cambodia-specific projects were proposed:

[3] Though the initial intent was for the Indian students to act as sub-contractors, they wanted to gain further experience in web-based software development. They consequently developed the whole software product in parallel with the software that was developed by the US students. One of the reasons why they were eager to do this was that they thought it would improve their resumes; reflecting their seriousness, they asked for certificates to present to future employers.

- **ITC Library Management System.** This project was to design and develop a system to replace the mainly paper-based activities of the ITC Library. The system had to support administrators, librarians and patrons, and provide the standard functionality of the existing system. An interesting aspect was to be uncovering the unique policies of the Cambodian library (i.e. the business logic). For example, there are no fees for the late return of books, as an honour system (preventing graduation) is in place.
- **Cambodian Crafts On-Line Store.** This project was to design and develop a system that would sell uniquely Cambodian crafts through the Internet.
- **Cambodian On-line Restaurant.** This project was to design and develop a virtual restaurant selling Cambodian dishes for home delivery via the Internet.

These last two E-Commerce projects were to manage the registration of customers, the placement of orders, and the fulfilment and control tasks of service staff.

2.5 Process, Technology and Communication Tools

Since this was a first software engineering class for the US students, they were exposed to a lightweight waterfall model with some iteration to help provide some overall shape and context. The software products developed by the US students were Java web-based applications (written using Servlets under Tomcat) with a back-end database implemented in MySQL or Oracle. Requirements mostly comprised textual documents and use cases, with some UML diagrams used for design. The development was all carried out within the Eclipse development environment. Students used the JUnit plug-in for unit testing and CVS for code sharing, and change and version management. Trac (*http://www.edgewall.com/trac*) open source wiki-based software was used for supply chain and project management. Clients used trac to report bugs, while developers used trac to fix and manage bugs. The Indian students produced Entity Relationship Diagrams for the database design; they did not use a particular tool. The Indian students developed their full software using JSP under Tomcat and Oracle. Students communicated using Yahoo! Groups mailing lists for asynchronous emails and using Yahoo! IM for synchronous chats. Local teams also communicated face-to-face. No specialized collaborative technology was used for distributed communication. Teams shared their work by posting document versions on their group websites.

3 Preparation

In this section, we describe the preparation that was necessary to set up a global supply chain management experience for students in terms of the project planning, communication coordination, faculty roles and continuous data gathering.

3.1 Project Planning

Many discussions on the countries, cultures, institutions, educational systems, academic calendars, students' background and Internet access had to take place to set up this project. Furthermore, the instructors had to design their syllabi (with grading policies) in collaboration and decide on the use of communication tools, CASE tools, and the software engineering process and communication protocols to be followed. Documents were exchanged between faculty and students, including country fact sheets, pictures of all students and faculty, and syllabi of individual courses. Instructors also needed to share all course materials (e.g. lecture notes, software engineering templates), exams, grades, feedback on the teams, video-taped presentations and software demonstrations for transparency. The milestones, schedules and deadlines of the projects for the three locations were designed in common and distributed as one document.

3.2 Communication and Coordination

The first important element that had to be taken into account in this study was the twelve hours time difference between Cambodia and the US, and the ten hours and thirty minutes time difference between India and the US. Another important element was Internet access. Cambodian students only had day-time access to the Internet from the ITC labs or from the widespread cyber-cafes for $1 an hour (an expensive proposition for Cambodians when the monthly average salary is around $60). Additionally, there were un-typically frequent power outages to work around due to the closure of an electricity power station in Phnom Penh. In one instance, this prevented the Cambodians from having access to the Internet for three days. By contrast, while Internet connectivity and power shortages were problematic at the university labs, most of the Indian students had reliable broadband high-speed access to the Internet from their homes. These factors meant that the US students and instructors had to coordinate and plan for communication. It was easier to contact the Cambodian students late at night and the Indian students early in the morning (US Eastern Standard Time).

3.3 Faculty and Their Roles

The faculty involved in this study have known each other for a long time. They have travelled intensively in Asia and in the US, and have a good understanding of US/India/Cambodia from a cultural, historical, educational, economical and political perspective, which is an advantageous foundation. Good relations and trust are crucial to the success of projects where there is a need for regular, open and transparent communication to plan, report, synchronize and solve problems in a timely manner. As part of the preparation, a site visit was made by one of the US professors to Cambodia and India for assessment of the infrastructure, coordination of the courses and syllabi, and to gauge the students' willingness and interest. The roles of the faculty had to be determined and agreed upon; faculty at one location had to oversee the three locations and play the role

of project manager [17]. The US faculty carried out this role and the other faculty reported to the project manager. It was also deemed necessary that only one professor handled these activities on a daily basis. Had there been more than one 'manager', there could have been problems of misunderstanding and miscommunication.

3.4 Data Gathering

Blogs and surveys were used to monitor and control aspects of the project relating to cultural differences, time and space complications, project activities and assessment of quality. The US students were active in using blogs to describe their work; the Cambodian and Indian students did not make use of blogs – they are not so widely used in their societies and the potential value is thus less clear.

- An entry survey was taken by all the students and the results shared to permit students to understand each other's background.
- The US teams answered a weekly questionnaire about their communications: type (e.g. emails, chats), scope (e.g. local, global), main topic (e.g. planning, feedback) and usefulness (e.g. high, medium, low). This questionnaire also recorded the reasons for any requirements changes (e.g. ambiguity, inconsistency, assumptions) and the instigating actors (i.e. Cambodia, India, US).
- A mid-semester survey was administered to all students to determine any logistical problems and to inform the study. This was also conducted to see how students perceived their team functioning (e.g. leadership effectiveness, balance of workloads, alignment of motivation, local versus global team biases, etc.)
- Individually, all students submitted post project statements on the overall experience and answered a post project survey that focused on what the students learned from each other, the issues and problems encountered, and the perceived effectiveness and usefulness of the experience.

4 Findings

Students were all positive with respect to their overall participation in this experience. In this section, we address the study questions we originally posed in Section 1 and organize our findings on the requirements engineering process, communication and coordination, social and cultural aspects, and interaction and quality[4].

4.1 Requirements Engineering

Requirements Engineering Process. The Cambodian and US students learned important lessons about requirements engineering in this setting: the

[4] The results in this section are derived from the post project surveys.

necessity of careful elicitation; the need for negotiation; why requirements descriptions should be unambiguous and well written; and the role of requirements validation to check understanding. Requirements were captured predominantly using questionnaires and clarified using (large) chat discussions. Requirements validation was achieved through (small) chats in the first instance and re-validated using checkbox documents where each requirement could be accepted, accepted with modification or rejected.

Requirements Changes. As clients, the Cambodian students recognized that they changed their mind on requirements quite regularly (56% agreement). Not surprisingly, the US developers perceived that the clients changed their minds frequently on the requirements (64%). Students appreciate how important it is to have a shared and aligned awareness on others' actions and responsiveness to avoid tensions, and they also see the realities of frequent requirements changes.

Requirements Assumptions. The projects demanded innovative and creative thinking because many of the assumptions that the US students had about what would constitute a feasible solution needed to be radically altered for the Cambodian market. For instance, in a country where Internet connectivity is slow and intermittent, students had to re-think the everyday model of E-Commerce (e.g. the use of ubiquitous graphics). The virtual shopping carts that many take for granted are unheard of in Cambodia and the metaphors do not always transfer across cultures. This situation forced students to differentiate between facts, constraints and assumptions, critical issues that often underpin many failed software development projects [12]. The Indian students made more assumptions about the Cambodian domain. This seems logical given the US students had direct contact with the client and the Indian students only gained information via the US intermediaries. However, the US students still made assumptions concerning the need to enforce the policy rules of the ITC library, making their system unusable in the client's eyes. Interestingly, the Cambodian students did not reject this unusable software and we suggest it is related to social bonding (discussed in Section 4.3) [7,8,13]. In the Indian version of the software, they added an additional late fee penalty. This addition was due to the lack of transfer of supplementary domain knowledge from the US students; they didn't consider the need to document how the honour system worked in their requirements. Also, the Indian students developed the craft store software such that it was restricted to ordering products from addresses within the US; this was not stated in the requirements document and suggests the kind of assumptions that can arise when the needs are passed on through a proxy, in this case it reflected assumptions about an American market more so than a Cambodian one.

4.2 Communication and Coordination

67% of the Indian students and 44% of the Cambodian students perceived coordination as the largest problem they faced when working with the US students. 25% of the Cambodian students perceived communication (i.e. language barriers) and the limited availability of the US developers as problematic. From the

Indian and Cambodian perspective, the main issue here was aligning themselves with the US students across time zones and busy schedules, especially since these students had almost twice as many hours of classes than the US students. The Cambodian students had a high class load, coupled with Internet access problems. The Indian students experienced similar issues, though actually cancelled chat meetings in the belief that the requirements document was written well enough for them to be able to develop the required software, preferring to write emails if they needed any clarification.

From the US perspective, 45% of the students perceived communication as the largest difficulty they encountered on the project. They would be offended when questioned about whether they were on target for meeting the milestones by the other students as they perceived this as questioning their ability to deliver. This could be attributed to cultural and/or language differences and might have been interpreted differently in a face-to-face situation. The US students, even though they had logistical difficulties in scheduling meetings, needed to be reminded constantly by the professors to be proactive. There would then be some frustration at the lengthy time the non-US students would take to respond.

The Cambodian students ranked good communications as a crucial factor for the success of a global software development project (56%). The Indian students ranked good communications and clear project plans as equally important (each 50%), reflecting their position as service providers who need to fit into a wider context and process. The US students emphasized good communications more than the others (76%), probably due to their direct experience of playing a coordinating role. While collaborative tools were ranked third by US students, they were not considered crucial by the Indian and Cambodian students. This may be partially due to less prior exposure to such tools and the fact that the US students were taking on the bridging role. Interestingly, both the Cambodian and US students thought that 'softer' skills were more important to the success of a global project than technical skills; the Indian students were more divided on this matter. Direct experience of the client/prime contractor relation likely motivates this 'soft skill' appreciation.

4.3 Social and Cultural Aspects

Relations. The interaction between the US and Indian students (as prime contractor and sub-contractor) was remarkably different from the interaction between the US and Cambodian students (as prime contractor and client). The former was abrupt and impersonal, focusing mostly on project matters. The latter was more polite and social (e.g. they all wanted to agree and seek understanding before moving on to the next topic). The topics discussed between the Indian and US students mainly revolved around the educational systems, their universities and home cities. Neither side felt they had expanded their knowledge of the other culture. In contrast, the Cambodian and US students discussed history, jobs and salaries, family structures and entertainment. They both felt they had increased their knowledge of each other's culture. These differences in inter-personal relations may be explained by the fact that the Indian and US

students already assumed they knew much about each other, whereas Cambodia is less well known. Also, the Indian students joined the project one month after the beginning of the project, so were really considered the 'hired help'. By contrast, the Cambodian students were the source of domain knowledge.

Learning from Each Other: US/India. The Indian students said they learned that US students "work according to the pre-specified plans and schedule and try to stick to those schedules. We in India don't go for the rigid plans." This seems in conflict with their reliance on good planning for successful projects, mentioned earlier in Section 4.2. When asked what they perceived the US students had learned about them, the Indian students said: "I think that would be determination and focus, which is a must for any successful professional." This emphasis on professionalism may be partially an explanation for expedience in communications. Further exemplifying this was the fact that once the US students had completed their final exams it was no longer possible to engage them further in the project. Their Indian counterparts, even after the end of their classes, still wanted to improve the software they had developed.

Learning from Each Other: Cambodia/US. The Cambodian students said that they had learned that US students "work until late at night". The US students said they learned that Cambodian students "do not have computers at home and go to school every day", "they are curious individuals who appreciate new technologies and educations", "they have much more school time than we do" and "they had no experience with credit cards and amazon.com". Credit cards and amazon.com usage may seem an integral part of life to US students, but they were not to the Cambodian students, and to some extent to the Indian students. The US students only discovered such realities during peripheral discussions with the Cambodians, not through project-specific questioning, and this was highly critical to the design of their E-Commerce systems. When asked what they believed was the most interesting thing the Cambodian students had learned from them, the US students said that they explained credit card usage and E-Commerce web site logic. They also added: "There seemed to be some assumption that all Americans are wealthy and opulent. I hope that our group served to dispel some of those myths."

Team Unity and Cohesion. Institution-specific gifts were exchanged between the US, Indian and Cambodian students at the beginning and end of the projects to create a community environment. In the mid-semester survey we found that the US students referred to the local team as 'the team', while the Cambodian and Indian students referred to the global team as 'the team'. When the Indian students discovered their names did not appear on the global team web site maintained by the US, this created some tension. Furthermore, the US students often referred to the Indian students as 'he', while five out of the six Indian students were female. The matter of gender amongst the Indian students is worthy of note and could be a possible reason for the nature of the terse communication outlined in Section 4.2. Indian girls may be more hesitant to instigate chat

conversation beyond what is required to complete a project task with an all male US team, asynchronous email being a more acceptable route.

4.4 Interaction and Quality

No statistical correlation was found between the quantity of interaction and the quality of the final product. However, it appears that the project with the most chats was ranked last by both the US and Cambodian faculty and rejected by its clients, while the project with the most emails was ranked first by all parties. Notably, the US software that was rejected by the client also used an early throw-away prototyping model to understand the requirements and discuss GUI options with the client. The project with the most explicit synchronous interaction with the client about requirements seemed to have the most problems, which (on first sight) conflicts with recognized best practice. The reality was that this local US group did not function as a team. While some of the students worked on a prototype with the clients, others in the team built a system to the original requirements, not accounting for the requirements learning. When the clients saw the product it not what was expected. They also ran into scheduling issues since they were keen to exploit technologies and tools that they did not have previous knowledge of. They became overwhelmed.

On the whole, the Cambodian students were more positive about the US software (the software implemented by the US with the Indian database component) than the purely Indian-built final software. Though technically limited in scope, it was ranked higher because it met the needs of the clients and did not make as many assumptions. However, the Indian software was actually more reliable in operation, and more care had been taken in creating a professional-looking product. One hypothesis could be that the friendly contacts between the US and Cambodian students led to a better experience and hence perception of the end product [7,8,13]. Another is the fact that the Indian students designed and built to second-hand information, and had no opportunity to manage the end clients' expectations throughout. Per the post project survey, the Indian students would have liked contact with the Cambodian students (83% expressed desire) and 68% of the Cambodian students concurred. Both parties believed this would have led to fewer assumptions and better products on the Indian side. Projects such as this do serve to convince students of the criticality of stakeholder contact and communications.

With respect to the sub-contracted work, the Indian students' designs were all very well done. However, assumptions were made with regard library late fees as discussed in Section 4.1, an example of incorrect information being passed down the supply chain. When the US team realized their error, they requested a change. Since the Indian team did not respond promptly, the US team went ahead and designed the database component themselves (still with some assumptions). This team included the most technically competent US students. It can be difficult to teach such students to delegate part of their work as this requires trust. Both the restaurant and craft store projects did use the Indian students' work, which integrated quite seamlessly. This was perhaps more possible since this technical knowledge was more lacking across these two US teams.

5 Conclusions and Recommendations

Software engineering education needs to reflect the realities of changing professional practice. Students should be prepared and flexible to adapt to whatever role they find themselves in. It is intrinsically feasible and practical for students to learn how to identify, build and integrate component parts of a software project in a multi-cultural university setting, and educationally critical for them to experience all sides. A global supply chain model provides a way for students to learn about the long-term skills that will be needed to augment their technical skills. In this study, two of the three projects were able to leverage skills in the global team that were lacking in the local team. However, care must be taken not to alienate those students taking on a sub-contractor role. While the relationship between the US/Cambodian students was carefully planned for by the instructors, the US/Indian relationship was left more to the students. Consequently, student photos were not distributed in a timely fashion and incorrect first impressions were made. Social bonding, getting to know the team members and their wider interests, lies at the heart of relationship management and has repercussions for the health of a global supply chain. Instructors need to instigate this process at the onset of the project and ensure mechanisms to facilitate this are built in throughout.

Regular communications are essential to ensure momentum and to keep projects on track. With supply chains, transparency is desirable all the way through the chain when students are learning about subtle dependencies and trust. Coupled with this is the important role of a shared process and an up-to-date version controlled repository for information exchange. Attention needs to be paid to explicitly designing a simple process and a straightforward communication and change management strategy with the students, and to avoid overwhelming them with the latest tools and technologies.

In a project of this nature, it is easy for course instructors to become project managers. Such a situation can prevent students from experiencing milestone setting, planning and coordination tasks. Projects need to be designed to offload more of this task on to the students so they understand its role in running a successful project and buy into it. One suggestion is to involve business or graduate software engineering students in this capacity. For a capstone course in software engineering, there is already much to cover.

The perception amongst the US students in the study was that they were doing the most work. An environment must be cultivated where the view is one of a shared venture in which all parties are contributing equally, albeit in different and equally valuable ways to prevent resentment. Issues of perception, along with trust, need as much attention as the technological and process skills that the course is teaching. More needs to be studied about the balance of competition and collaboration on global student projects of this kind. The Indian students wanted to create their own software and desired technological perfection. Relations were a little smoother between those students with whom the US students did not feel they were competing.

While some global team working skills can be learned on the job, others may need to be explicitly taught in the software engineering classroom. The global teams that operated the most effectively were the ones where the local teams (the ones playing the integrative role), functioned well internally and had the process in place to broaden out.

Acknowledgments. This project was undertaken under the auspices of a Pace University presidential grant. We would like to thank all the students who participated. We are grateful to Professors Sok and Chanthearith who helped guide the interactions between the Cambodian and US students. We thank Doug Tidwell, IBM cyber-evangelist, for the Eclipse training he provided to the US students. Finally, we thank Dean Susan Merritt for her support with this project.

References

1. Aspray, W., Mayadas, A.F., Vardi, M.Y: Educational Response to Offshore Outsourcing. In: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (2006), March 3-5, pp. 330–331. Houston, Texa, USA (2006)
2. Aspray, W., Mayadas, F., Vardi, M.Y: Globalization and Offshoring of Software. A Report of the ACM Job Migration Task Force (2006)
3. Audy, J., Evaristo, R., Watson-Manheim, M.B.: Distributed Analysis: The Last Frontier? In: Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS'04), p. 10010. Big Island, Hawaii (2004)
4. Coar, K.: The Sun Never Sets on Distributed Development. ACM Queue 1(9), 32–39 (2004)
5. Crosby, P.B.: Quality Is Free: The Art of Making QualityvCertain. McGraw Hill, New York (1979)
6. Damian, D., Hadwin, A., Al-Ani, B.: Instructional Design and Assessment Strategies for Teaching Global Software Development: A Framework. In: Proceedings of the 28th International Conference on Software Engineering (ICSE'06), May 20-28, pp. 685–690. Shanghai, China (2006)
7. Damian, D., Zowghi, D.: Requirements Engineering Challenges in Multi-site Software Development Organizations. Requirements Engineering Journal 8(1), 149–160 (2003)
8. Favela, J., Pe-Mora, F.: An Experience in Collaborative Software Engineering Education. IEEE Software 18(2), 47–53 (2001)
9. Ferguson, E., Henderson, P., Huen, W., Kussmaul, C.: IT Offshore Outsourcing: Impact on CS/IS Curriculum. In: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, February 23-27, pp. 258–259. St. Louis, Missouri, USA (2005)
10. Ferguson, E., Kussmaul, C., McCracken, D., Robbert, M.A.: Offshore Outsourcing: Current Conditions and Diagnosis. In: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, March 3-7, pp. 330–331. Norfolk, Virginia, USA (2004)
11. Gotel, O., Scharff, C., Seng, S.: Preparing Computer Science Students for Global Software Development. In: Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference (FIE'06), San Diego, California (2006)
12. Jackson, M.: Software Requirements and Specifications. Addison-Wesley Professional (1995)

13. Kobylinski, R., Creighton, O., Dutoit, A., Bruegge, B.: Building Awareness in Global Software Engineering: Using Issues as Context. In: Proceedings of the International Workshop on Distributed Software Development (GSD'02), Orlando, Florida, May 21 (2002)
14. McCracken, W.M.: Counter Point-SE Education: What Academia Can Do. IEEE Software 14(6), 27–29 (1997)
15. Meyer, B.: The Unspoken Revolution in Software Engineering. IEEE Computer 39(1), 121–123 (2006)
16. Olson, J.S., Olson, G.M.: Culture Surprises in Remote Software Development Teams. ACM Queue 1(9), 52–59 (2004)
17. Petkovic, D., Thompson, G., Todtenhoefer, R.: 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 (ITiCSE'06), June 26-28, pp. 294–298. Bologna, Italy (2006)
18. Purvis, M., Purvis, M., Cranefield, S.: Educational Experiences from a Global Software Engineering (GSE) Project. In: Proceedings of the 6th Conference on Australasian Computing Education (ACE'04), pp. 269–275. Dunedin, New Zealand (2004)
19. Ribeiro, J.: Indian Outsourcers Continue to Make Gains. Computerworld (August 14 2006)
20. Richardson, I., Milewski, A.E., Mullick, N., Keil, P.: Distributed Development: An Education Perspective on the Global Studio Project. In: Proceedings of the 28th International Conference on Software Engineering (ICSE'06), Shanghai, China, May 20 - 28, pp. 679–684 (2006)
21. Tromby, M., Marcus, B.: Bridging the Chinese Skills Gap. Computerworld (June 6 2006)
22. Xiaoqing, L.: Collaborative Global Software Development and Education. In: Proceedings of the 29th International Computer Software and Applications Conference (COMPSAC'05), p. 371. Edinburgh, Scotland (2005)