Integration Starts on Day One in Global Software Development Projects

Christelle Scharff,
(cscharff@pace.edu)
Olly Gotel, Vidya Kulkarni, Longchrea Neak
(ogotel@pace.edu, vidya_kulkarni@hotmail.com, longchrea@gmail.com)
http://atlantis.seidenberg.pace.edu/wiki/gsd2007

---ICGSE 2008---
Outline

- Motivation
- Background
- Project description and setting for 2007
- Global project teams, their process and tools
- Emphasis on integration at different levels
  - Sub-components (including third-party supplied components)
  - Future requirements
  - Social integration
- Findings
- Lessons and recommendations
- Summary and sneak preview of 2008
Motivation

- 4 year initiative to date
- Balanced first-hand view of Offshore Outsourcing:
  - What roles will students play in a global market place?
  - What opportunities could arise for them?
- Exposure to realities of global supply chain management:
  - How to divide up a project into component parts for different parties to work on across time zones and cultures?
  - How to deal with RFPs / competition in supply?
  - How to integrate work and deploy into a market?
  - How to control and assure quality in distributed projects?
- Understand BOTH technical and ‘softer’ skills:
  - What process and communication models work?
  - What blend of technology can be used?
  - Accounting for a diverse playing field?
  - A model for partnering, auditing and mentorship?
2 years of Global Software Development Projects:

- Pace (www.pace.edu) and ITC (www.itc.edu.kh), then UoD (www.du.ac.in)
- Model for running student GSD projects integrated within the curricula
- Established relations between professors - Exchanges and visits

Students:

- Undergraduate Software Engineering at Pace and ITC
- Graduate Software Quality Assurance at Pace
- Graduate Database Applications at UoD

Particular issues and challenges to focus on in 2007:

- To complete one project – Focus on integration issues
- Process and technology
- Quality matters - Included Graduate Software Quality Assurance students from Pace

2008 now complete - Extended quality focus and competition!
2007 Project

- **MultiLIB** - ITC Computer Science Department Library Management System:
  - Currently paper-based
  - Support students, librarians, professors and administrators
  - Enforce the unique library policy
  - Maintain a holding of different types of resources (e.g., books, CD-ROMs, E-books, videos and students’ reports)

- 3 sub-components:
  - Student / professor / guest side
  - Librarian / administrator side
  - Innovation
  - With underlying database…
Setting for 2007

Globalization

Software engineering

USA
Managers
Developers
Auditors
Mentors

Cambodia
Clients
Testers

India
Sub-Contractors

Communications

Mentoring and auditing

Technology

RFP

12 hours time difference

9.5 hours time difference

13
Roles and Responsibilities

- **Cambodian undergraduate students** – Clients, testers and translators:
  - Describe environment / problem / software
  - Review and give feedback on requirements, design and testing documents
  - Test the software and submit bug reports
  - Report on the Pace team
  - Accept or reject the software
  - Compare the software developed in the US and in India
  - Present their experience
  - Demonstrate the software
  - Implement the software
Roles and Responsibilities

- **Undergraduate US students** – Developers and lead contractors:
  - “Capture” the requirements,
  - Propose design options
  - Manage an RFP process
  - Implement and test the software
  - Handle requirements changes and integrate feedback
  - Deliver software for their client
  - Report on the ITC / Delhi team
  - Describe and reflect on the software engineering process and communication protocol followed
  - Present their work professionally
  - Demonstrate their software
Roles and Responsibilities

- **Indian graduate students** – Third-party suppliers:
  - Answer an RFP
  - Provide US students with a database design and SQL code to be integrated into the overall system design
  - Report on the US team
  - *Implement the software*
Roles and Responsibilities

- **Graduate US students** – Quality mentors and auditors:
  - **Mentors (also IT professionals):**
    - One graduate advisor assigned to each US sub-team
    - One graduate advisor assigned to overall team integration
    - Weekly support structure
    - Coaching with techniques and practices
    - “The offering of advice, information, or guidance by a person with useful experience, skills, or expertise for another individual’s personal and professional development.” [HBS]
  - **Auditors (one in each pair also an IT professional):**
    - Two graduates assigned to audit each US sub-team
    - Provide quality checklists for audit
    - Review and report on student work (each deliverable and process)
    - External and independent quality gate keepers

*Plan, log, report*
# Global Project Teams

<table>
<thead>
<tr>
<th>Country</th>
<th>Student Side</th>
<th>Librarian Side</th>
<th>Innovation Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cambodia</td>
<td>4 clients</td>
<td>5 clients / testers</td>
<td>4 in all roles</td>
</tr>
<tr>
<td>US UGs</td>
<td>4 developers</td>
<td>4 developers</td>
<td>--</td>
</tr>
<tr>
<td>US Grads</td>
<td>1.5 mentors</td>
<td>1.5 mentors</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>2 auditors</td>
<td>2 auditors</td>
<td>--</td>
</tr>
<tr>
<td>India</td>
<td>3 teams of 2 sub-contactors</td>
<td>2 sub-contactors</td>
<td>--</td>
</tr>
</tbody>
</table>
Process

- Set-up
  - Initialization of communications and team bonding with gifts and videos (1 week)
- Requirements (5 weeks)
- Design (4 weeks)
- Mid-semester presentations
- Implementation (3 weeks)
- Testing (1 week)
- Final presentations
- Through-life (all 14 weeks):
  - Project management and communications
  - Feedback and iteration (via mentoring and auditing)
  - Socialization
<table>
<thead>
<tr>
<th>Activity</th>
<th>Tool</th>
<th>Rationale and tasks supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>MS Word, Chats, Email</td>
<td>Requirements gathering, elicitation and validation. Synchronous communication. Asynchronous communication.</td>
</tr>
<tr>
<td>Design</td>
<td>SmartDraw UML or similar</td>
<td>To model design options and achieve a better understanding of how the system should behave and correspond to client needs; To facilitate communication with database designers in India; To produce an ERD.</td>
</tr>
<tr>
<td>Implementation</td>
<td>Eclipse, Java/JSP, Apache Tomcat, MySQL</td>
<td>To take advantage of the IDE features, JUnit and Subversion plugins, and the externalization mechanism. To encourage students to build on their Java / JSP skills. To use an open source servlet container that would be easily deployable in Cambodia. To use an open source DBMS that would be easily deployable in Cambodia.</td>
</tr>
<tr>
<td>Testing</td>
<td>JUnit, Java.net</td>
<td>To automatically run unit tests and validate units of the software. To validate software; clients and developers used the Issue Tracker facility of java.net to report, then fix and manage bugs respectively.</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Subversion</td>
<td>To contain all documents and software artifacts; To increase milestone visibility and awareness; To agree on shared architecture, approve database design, gain feedback on user interface mockups, clarify deployment environment, communicate responsibility. To allow problems to be addressed by the teams and instructors.</td>
</tr>
<tr>
<td>Project Management</td>
<td>Wiki, Blog</td>
<td>To facilitate code sharing, change and version management.</td>
</tr>
<tr>
<td>RFP Process</td>
<td>MS Word, Email, Chat</td>
<td>To write RFP and acceptance / rejection letters with justifications. To solicit bids, manage RFP process and select winning design. To clarify RFP requirements.</td>
</tr>
<tr>
<td>Socialization</td>
<td>Video, Chat</td>
<td>To get to know each other and put a face on a name. To get to know each other and facilitate spontaneous conversation.</td>
</tr>
<tr>
<td>Mentoring</td>
<td>Face-to-face</td>
<td>To provide technical and team management assistance.</td>
</tr>
<tr>
<td>Auditing</td>
<td>Wiki, Face-to-face</td>
<td>To review artifacts produced and check compliance. To conduct interviews.</td>
</tr>
<tr>
<td>Instructor Oversight</td>
<td>Wiki, Blog, Survey</td>
<td>To monitor progress and deliverables. To address problems and have students elaborate on particular questions asked by instructors for assessment purposes. To gather students’ perception on project, about RFP process, mentor / auditor experience and overall experience.</td>
</tr>
</tbody>
</table>
To Emphasize Integration...

- **Integration of the two core sides** – Need of the student / professor / guest and librarian / administrator sides to fuse their work and champion their respected sides inside an integrated framework.

- **Integration of a third-party supplied component** – Three Indian sub-teams answered an RFP process requiring a database design to be integrated in MultiLIB by the US students.

- **Integration of future requirements** – The Cambodian innovation sub-team was to explore and identify future requirements for MultiLIB to permit their integration in the current requirements and design.

- **Social integration** – The social side and team cohesion of the project was emphasized via exchange of country-specific gifts and videos about students’ lives between the US and Cambodian students particularly.
Findings: Integration

- Integration needs to be planned early
  - US undergraduates started the project as two local sub-teams and worked on the requirements independently, which resulted in two requirements documents – even for the common parts - and little trust in the quality of the other sub-team’s work in the remainder of the project.

- Documentation to facilitate integration
  - The database sub-component that was selected and convinced the US students was “well-organized, contained the required MySQL code, and was well-documented”.

- Technology to support integration
  - The US students used two separate wikis – one for each core side of MultiLIB – which created communication silos. A shared wiki came too late in the course of the project to unify the US team.

Not a failure in the concept
Findings: Integration

- Diverse design options to threaten integration
  - The US students produced their own database design based on the diverse options provided by the Indian students. This resulted in passionate comments from the Indian students: “I would like to know why they did not integrate our database design as they themselves found it the best”; “This is quite unprofessional and moreover if they had any issues then we should have been made aware of it so that we could have given our support”; “It is difficult to understand the design of other students. But it is a kind of violation of the whole concept of global software development”.

- Communication to impact integration
  - Communication bottlenecks resulted in integration risks for the Indian students who experienced delays in getting answers from the client because questions were mediated through the US students and some doubts were never clarified.
Findings: Integration

- Team leadership to direct integration
  - Integration was carried out by the leaders of each core side of MultiLIB. An integration leader would have been necessary to erase the initial sub-divisions of the work and in the teams.

- Socialization to foster team integration
  - The way the students were introduced to the project impacted team cohesion – the six Indian students organized themselves as a unique team who worked in a shared purpose but the US students never reached that level of collaboration.
  - Integration to account the work of the Cambodian innovation team was narrowed by all the other integration aspects. The role of the innovation team should have been more emphasized.
  - Students considered themselves more as local teams than a global team. Socialization efforts between all partners should have been sustained to encourage social integration.
# Integration: Dos and Don’ts

<table>
<thead>
<tr>
<th><strong>DO</strong></th>
<th><strong>DON’T</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan for integration from day one, or preferably much earlier.</td>
<td>Underestimate integration issues and think they can be addressed on the fly.</td>
</tr>
<tr>
<td>Focus on the team as a whole, its goal and motivations, and the overall architecture of the system before dividing work.</td>
<td>Divide the work without first understanding the larger team and environment in which these contributions will play a role.</td>
</tr>
<tr>
<td>Invest on socialization for team cohesion.</td>
<td>Tolerate disrespect based on ignorance and differences, and don’t expect this necessary learning to happen without support and time built in for it.</td>
</tr>
<tr>
<td>Ensure awareness by agreeing and establishing protocols for communication.</td>
<td>Rely upon pull-driven approaches and communication silos.</td>
</tr>
<tr>
<td>Publicize and remind the role of everybody on the project.</td>
<td>Isolate anybody by failing to explain the rationale for changed decisions and ignoring emotions.</td>
</tr>
</tbody>
</table>
# Integration: Dos and Don’ts

<table>
<thead>
<tr>
<th><strong>DO</strong></th>
<th><strong>DON’T</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use a common set of consensual and available technologies across locations.</td>
<td>Impose technologies without accounting for local perceptions, constraints and training needs.</td>
</tr>
<tr>
<td>Use a process that sustains involvement of the client and allocates time for feedback.</td>
<td>Focus on the pressures and concerns of the local team in isolation of the bigger picture.</td>
</tr>
<tr>
<td>Create a trustful environment supporting work delegation where respect can be earned.</td>
<td>Establish a context in which one person or team is encouraged to believe they know best.</td>
</tr>
<tr>
<td>Have integration leaders for development, communication and socialization.</td>
<td>Misjudge the importance of integration leadership at socio-technical and disparate levels.</td>
</tr>
<tr>
<td>Clearly and concisely document any sub-components, interfaces and integration processes.</td>
<td>Allow teams to run blind without documentation of any to-be integrated artifacts.</td>
</tr>
</tbody>
</table>
Summary

- We have presented some of the factors that serve to either support or jeopardize the integration process in the context of global software development.
- GSD projects are complex beasts, demand management overhead and can be subject to quality problems - No wonder not many academic institutions teach and prepare for this
- Models for teaching GSD need to be sustainable and scalable - Even more disincentive to run these projects in schools
- We have been evolving a model for international undergraduate and graduate students to work together as partners and mentors to address some of these challenges - It is an approach to institutional learning that benefits all sides and extends the reach of the classroom
- From an overworked instructor to networks of support at many levels - Quality (internal and external) is the perfect focus for these networks - It DOES scale up and does improve quality

Sneak Preview of 2008...
Technology

Deployment

Communications

Mentoring and auditing

Socialization

http://atlantis.seidenberg.pace.edu/wiki/gsd2008
Client quality coaches (5 US graduates)

to help the client to baseline the requirements, create a versioning and requirements management process, and help prepare for acceptance testing and software selection.

Client (5 Cambodian ITC students)

to manage the requirements and maintain a requirements wiki, and each student sponsors a development team.

5 development teams
5 versions of the software!

US NYC (6 students)

coaching

Sponsors

US PLV (4 students)

socializing

Socialization team (2 Cambodian RUPP students)
to help with socialization.

Thailand (4 students)

training

SQA trainees (4 Thai students)
to shadow/learn from the coaches and auditors.

India (5 students)

coaching

Auditors (16 US graduates and IT professionals)
to provide early feedback on the requirements, audit each development team and externally test the software delivered. One student is the SQA Manager.

Cambodia ITC (4 students)

coaching

5 development teams
5 versions of the software!

Developer quality coaches (5 US graduates)
to help each development team to inject quality into their process and products.

US NYC (6 students)

coaching

Developer quality coaches (5 US graduates)
to help each development team to inject quality into their process and products.

US PLV (4 students)

socializing

Socialization team (2 Cambodian RUPP students)
to help with socialization.

Thailand (4 students)

training

SQA trainees (4 Thai students)
to shadow/learn from the coaches and auditors.

India (5 students)

coaching

Auditors (16 US graduates and IT professionals)
to provide early feedback on the requirements, audit each development team and externally test the software delivered. One student is the SQA Manager.

Cambodia ITC (4 students)

coaching

5 development teams
5 versions of the software!

Project with SQA (14+ weeks)

Code/Test (6)

Deploy/Maintain (n)

Requirements (4)

Design/Prototype (4)

Bonding/Setup (2)

Framework for the Software Development Lifecycle - feedback and iteration triggered by coaching and auditing
Thanks

- Students (ITC, Pace, Delhi)
- Faculty (ITC, Pace, Delhi)
- David Michael, United Business Media, New York
- Cedric Mainguy, Asiaform, Phnom Penh
- John Fox, Homeland Energy Inc., New York
- Doug Tidwell, IBM Cyber-evangelist
- Chris Nelson, IBM Senior Software Engineer
- Gary Thompson, Sun Microsystems
- Guillaume Maillard, ILM informatique, France
- NCIIA Grant 2006-2008
- Seidenberg School of CSIS, Pace University

http://atlantis.seidenberg.pace.edu/wiki/gsd2007
http://atlantis.seidenberg.pace.edu/wiki/gsd2008