Evolving an Infrastructure for Student Global Software Development Projects: Lessons for Industry

Olly Gotel (Pace University, USA)
Vidya Kulkarni (University of Delhi, India)
Des Phal (Royal University of Phnom Penh, Cambodia)
Moniphal Say (Institute of Technology of Cambodia, Cambodia)
Christelle Scharff (Pace University, USA)
Thanwadee Sunetnanta (Mahidol University, Thailand)

ISEC 2009 - India
Outline

- IT Offshore Outsourcing
- Global Software Development and its impact on CS education
- Motivations
- Tools for projects 2005 onwards
- GSD Project set up-2008 teams and responsibilities
- Tool for Software Development Process
- Findings
  - Use of tools by teams
  - Perception of tools by teams
- Lessons and recommendation
Global Software Development

- When software is developed offshore at different places, it is called Global Software Development

- Teams are distributed geographically

  - Different cultures,
  - Different languages,
  - Different time zones

→ Challenges !!
Despite of the challenges, many companies have adopted the GSD because of:

- Reduced Development Cost in remote places
- Availability of high qualified people in country like India, China because of advances in CS education
- Advances in ICT that has facilitated the collaboration, and communication that are essential for GSD
GSD and its Impact on Computer Science Education

- Entry Level Jobs migrating to Service-Providing Countries
- Decline in CS enrollment in US due to fear of not finding jobs after graduation
- Increase in CS enrollment in India and China
- Opening of many new IT Institutes in India
- Educational efforts to teach GSD in classrooms through projects with the involvement of other educational institutes in other countries
Motivations

- 4 years initiative to date starting from 2005 onwards

- 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 integrate work and deploy into a market?
  - How to control and assure quality in distributed projects?
Motivations (Cont.)

- Understand BOTH technical and “softer“ skills:
  - What process and communication models work?
  - What blend of technology can be used?
  - A model for partnering, auditing and mentorship
First Step - 2005

<table>
<thead>
<tr>
<th>Tool Type</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical</td>
<td>Eclipse IDE with JUnit, MySQL</td>
</tr>
<tr>
<td>Communication</td>
<td>Yahoo IM, Yahoo groups, Websites, blogs</td>
</tr>
<tr>
<td>Socialization</td>
<td>Yahoo IM, exchange of gifts</td>
</tr>
<tr>
<td>Project Management</td>
<td>Websites, pages, blogs</td>
</tr>
</tbody>
</table>

**Managers / Developers**

**Clients**

**Globalization**
3rd Step - 2007

<table>
<thead>
<tr>
<th>Tool type</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical</td>
<td>Eclipse IDE with JUnit and subversion, MySQL, javanet for issue tracking</td>
</tr>
<tr>
<td>Communication</td>
<td>Yahoo IM, Yahoo groups, wikis, blogs, videos</td>
</tr>
<tr>
<td>Socialization</td>
<td>Videos, Yahoo IM, exchange of gifts, videos</td>
</tr>
<tr>
<td>Project management</td>
<td>Wikis, blogs</td>
</tr>
</tbody>
</table>

USA
- Managers
- Developers
- Auditors
- Mentors

CAMBODIA
- Auditors
- Mentors

INDIA
- Sub-Contractors
- Clients
- Testers

Globalization

Socialization

Technology

Software engineering process
4th Step - 2008

Globalization
Socialization
Competition
Technology
Software Engineering

12 hours

US
Pace University
NYC Campus

9.5 hours

US
Pace University
Pleasantville Campus

INDIA
University of Delhi

1.5 hours

CAMBODIA
Royal University of Phnom Penh

THAILAND
Mahidol University

CAMBODIA
Institute of Technology of Cambodia

US Students and IT Professionals (Global Bank in NYC)

SQA 2008

QuickTime® and a TIFF (Uncompressed) decompressor are needed to see this picture.
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.

5 development teams 5 versions of the software!

Client (5 Cambodian ITC students) to manage the requirements and maintain a requirements wiki, and each student sponsors a development team.

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

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

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

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.

Framework for the Software Development Lifecycle - feedback and iteration triggered by coaching and auditing.
<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, Wikis</td>
<td>Requirements gathering, elicitation and validation. Synchronous communication. Asynchronous communication. Requirements documents (versioned and drafts), FAQs.</td>
</tr>
<tr>
<td>Design</td>
<td>SmartDraw, Eclipse, DB Designer</td>
<td>To model design options and achieve a better understanding of how the system should behave and correspond to client needs. To take advantage of the IDE features, JUnit and Subversion plugins, and the externalization mechanism.</td>
</tr>
<tr>
<td></td>
<td>Java/SP, Apache Tomcat, MySQL, Netbeans/Visual Studio</td>
<td>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. Asian and Thai choices.</td>
</tr>
<tr>
<td></td>
<td>Unit, Wink, Chat</td>
<td>To automatically run unit tests and validate units of the software. For demos. For code walkthroughs. 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>Testing</td>
<td>C#/ASP.NET, JUnit, Java.net</td>
<td>To facilitate code sharing, change and version management. 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 responsibilities.</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Subversion</td>
<td>To disseminate checklists, access artifacts produced and post audits. Communication with team.</td>
</tr>
<tr>
<td>Project Management</td>
<td>Wikis*, Timezone Software, Google Calendars, Mailing Lists, Wiki's*</td>
<td>Social experiment. Trigger understanding and learning. Putting faces to names for extended teams. To get to know each other and facilitate spontaneous conversation. To provide technical and team management assistance.</td>
</tr>
<tr>
<td>Socialization</td>
<td>Second Life Exercises, Photos, Chat, Face-to-face</td>
<td>Training and guidance. Create and share useful resources. Communication with team.</td>
</tr>
<tr>
<td>Coaching</td>
<td>Members of the developers, Face-to-face, chat, mailing lists</td>
<td>To monitor progress and deliverables. To address problems and have students elaborate on particular questions asked by instructors for assessment purposes. Regular class meetings and scrum sessions with teams.</td>
</tr>
<tr>
<td>Auditing</td>
<td>Wikis, Face-to-face, chat, mailing lists</td>
<td>To gather students' perception on project, about RFP process, mentor / auditor experience and overall experience.</td>
</tr>
<tr>
<td>Instructor Oversight</td>
<td>Mailing Lists, Face-to-face, Survey</td>
<td>To disseminate checklists, access artifacts produced and post audits. Communication with team.</td>
</tr>
</tbody>
</table>
TOOLS

Communication:
- Yahoo Messenger: chat
- Yahoo groups: for mailing list
- Emails

Engineering
- Language: Java, JSP and ASP.net, C# for Thai teams
- IDE: Eclipse, NetBeans
- Design: StarUML, DB4 Designer
- Issue tracking: Java.net
- Subversion:
- Web server: Apache Tomcat, IIS
- Database server: MySQL and LDAP
- Testing: JUnit
• Project Management:
  • Wiki
  • Google Calendar
  • Time Zone Software
  • Yahoo groups: for mailing list
  • Emails

• Socialization
  • Second life
  • Yahoo messenger
  • Map and Face exercises
<table>
<thead>
<tr>
<th>Team</th>
<th>What were the new technologies?</th>
<th>Most difficult technology to learn?</th>
<th>Killer technical tool?</th>
<th>Killer communication tool?</th>
<th>Two most crucial tools (overall)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developers - Cambodia</td>
<td>For all teams: their IDE, design tools, subversion, Java.net issue tracker, wiki</td>
<td>Java.net</td>
<td>IDE</td>
<td>Yahoo IM</td>
<td>IDE / Emails</td>
</tr>
<tr>
<td>Developers - India</td>
<td>IDE</td>
<td>IDE</td>
<td>Emails</td>
<td>IDE / Java.net issue tracker</td>
<td></td>
</tr>
<tr>
<td>Developers - Thailand</td>
<td>IDE</td>
<td>IDE</td>
<td>Emails</td>
<td>IDE / Emails</td>
<td></td>
</tr>
<tr>
<td>Developers - US NYC</td>
<td>Design tools</td>
<td>IDE</td>
<td>Yahoo IM</td>
<td>Emails / IDE</td>
<td></td>
</tr>
<tr>
<td>Developers - US PLV</td>
<td>Design tools</td>
<td>IDE</td>
<td>Emails</td>
<td>IDE / Emails</td>
<td></td>
</tr>
</tbody>
</table>
Lessons

- Incredible amount of work to set-up and maintain
- Need to provide tutorials and hands-on labs even for the simplest of tools (uneven learning curve)
- While you need redundancy in communication channels - too many wikis and mailing lists overwhelm the students
- Servers WILL fail - plan for it
- Make sure there is something for everybody
- Establish a good relation with a friendly Internet café
- Understand what is local and what is global -- a good framework should support local technology preferences
RECOMMENDATIONS

- **Time zones**, find the reasonable overlaps for live discussions as asynchronous communication alone can lead to failure.

- **Synchronicity**, make sure you have way to get your IM works and try to pick a system that works for multi-way conversation.

- **Everyday awareness of simple things**, make sure you keep all parties in the project informed about the others job titles, vacations, email and their preferred contact mode. Wiki helped us on this.

- **Collaboration / coordination tools**, using web-based collaboration (e.g., a wiki) allows people to share documents across geographies.
RECOMMENDATIONS

- **Time to learn other development tools**, it takes time to learn new tools, so try to keep this in the schedule too.

- **Software versions**, the incompatibility between different versions of software should be preventable (e.g., office 2000/2003/2007)

- **Virtual social gatherings**, find the way for all parties involved to meet, if not in person, find alternative way, this help establish bonds between them.

- **Have a trusted cornerstone to the project - visit your offshore team**, try to make a visit, to get to know people, explain the context of the project, as well as to build trusted relationships.
For More Information…

See our GSD2008 wiki:
http://atlantis.seidenberg.pace.edu/wiki/gsd2008

… and contact me or any of the global professors
Acknowledgment

This work is supported by a National Collegiate Inventors and Innovators Alliance grant (#3465-06) -- “Incubating the Next Generation of Global Software Development Entrepreneurs” -- (2006-2008) and a Campus Second Life scholarship.

We thank all the 159 (60 in 2008) students who have been involved to date and the many ITC faculty.
Conclusion

- Our GSD model has been evolved over a period of four years using the lessons learnt from previous years.

- In future, we plan to include project management students who can help the instructors.

- Finally, all students enjoyed working on real-life project in real industry-like environment.
Thank you for your kind attention
**Offshore Outsourcing**

- **Outsourcing** is the delegation of tasks or jobs from internal production to an external entity (such as a subcontractor). [Wikipedia]

- **Offshoring** can be defined as relocation of business processes (including production/manufacturing) to a lower cost location, usually overseas. [Wikipedia]

- **Offshore Outsourcing** is the practice of hiring an external organization to perform some or all business functions in a country other than the one where the product will be sold or consumed. [Wikipedia]
# Uses of tools per teams

<table>
<thead>
<tr>
<th>Tool</th>
<th>Client</th>
<th>Client coach</th>
<th>Cambodia</th>
<th>India</th>
<th>Thailand</th>
<th>US NYC</th>
<th>US PLV</th>
<th>Developer coach</th>
<th>Auditors</th>
<th>Socialization team</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS Word</td>
<td>H</td>
<td>H R</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H R</td>
<td></td>
</tr>
<tr>
<td>UML / DB Design</td>
<td></td>
<td></td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td>R</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>IDE</td>
<td>D</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>R</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td>D</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>R</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Java / JSP / C#</td>
<td>D</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>R</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Sub-version</td>
<td>D</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>JUnit / Nunit</td>
<td></td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Java.net</td>
<td></td>
<td></td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>M</td>
<td>H</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>L</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Chat</td>
<td>M</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>L</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Wiki</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Demo Video</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Second Life</td>
<td>L (P)</td>
<td>H (P)</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>