Transitioning to Distributed Development in Students’ Global Software Development Projects: The Role of Agile Methodologies and End-to-End Tooling

Christelle Scharff
Pace University, NY, USA

Dr. Olly Gotel
Independent Researcher, USA

Prof. Vidya Kulkarni
University of Delhi, India

Thanks:
NCIIA
IBM
Agenda

- Global Software Development (GSD) 2005-2008
- Focus for 2009
- Research questions
- GSD 2009 setting
- What is Scrum?
- Scrum implementation
- IBM Rational Team Concert (RTC)
- Project outcomes
- Findings
- Sneak pick of 2010
Local Team Spirit / Mashups of Tools / Want to be a Developer / Late First Working Version of a Software
Local Team Spirit / Mashups of Tools / Want to be a Developer / Late First Working Version of a Software
Research Questions

- **Role of the Process** -- How well do Agile and Scrum practices support the work of distributed developers?

- **Role of the Tooling** -- How important is tooling in supporting distributed developers using Agile and Scrum practices?

- **Guidelines** -- How are Agile and Scrum practices best introduced into distributed students’ projects?
GSD 2009 – Project Setting

- Distributed Developers
- End-to-Eng Tooling
- Scrum
- Mobile
- Agile Methodologies

- USA (1)
- Senegal (2)
- India (2)
- Developers (5)
- Process coach (1)
- Rotating Scrum Masters (each team)
- Product owner
Software Development Project

- **TargetFirstGrade – Product owner: Dr. Scharff**
  - Mobile application to assess the learning of pupils in first grade (5-6 year old) in topics such as Mathematics, Reading, Writing and Geography
  - To be used in large classes in the developing world
  - Delivery of exercises in these topics in the form of open-ended and multiple choice questions
  - Automated computation of the scores
  - SMS of the scores to the teachers and parents
  - Customization of the list of topics and problems by the teachers
  - English and French versions

*The Product Backlog of Target First Grade comprised 45 user stories – 18 high, 16 medium and 11 low priority user stories.*
What is Scrum?
Scrum

- Developed in management in 1983 and adapted to software development in 1993 by Jeff Sutherland and Ken Schwaber
- Empirical challenges cannot be addressed successfully by generic models
- Focus on maximizing the team’s ability in an agile manner to emerging challenges
- No specific process prescribed but often used with Agile
- Short iterations (Sprint) where the software is designed, developed and tested by the Scrum Team
- Daily 30-minute stand-up meetings (Scrum) to answer 3 questions
- The requirements are initially expressed using user stories and available in the Product Backlog and then in the Sprint Backlog
- The Product Owner is the owner of the requirements
- The Scrum Master facilitates Scrum and remove impediments linked with the process
Overview of Scrum

- Sprint Backlog
- Daily Scrum Meeting
- Backlog tasks expanded by team
- 30 days
- Potentially Shippable Product Increment

Source: Agile Software Development with Scrum by Ken Schwaber and Mike Beedle, and Cohn

Source: http://www.rallydev.com
Scrum Implementation

- **Communicate.** Sharing information creates visibility, better decision-making and a common understanding of shared goals.

- **Empower the team.** Nothing is more powerful than a team that is in control of its own destiny – a team that thinks the only thing limiting what they can accomplish is how creative they are and how hard they work.

- **Learn and improve.** Learning is about trying something, looking at the results and then improving.

- **Deliver value early.** Build trust with people by prioritizing work, committing to deliverables and delivering them reliably.
Scrum Implementation

- Distributed Developers
- End-to-Eng Tooling
- Scrum
- Mobile
- Agile Methodologies

- USA (1)
- Senegal (2)
- India (2)

- Developers (5)
- Process coach (1)
- Rotating Scrum Masters (each team)
- Product owner

- Time:
  - 5h30
  - 10h30
  - 5h30

- Locations:
  - USA
  - Senegal
  - India
Initialization and training (3)

Sprint planning
Scrum of Scrums / Process coach / feedback
Sprint Demo / Retrospective / Planning / Process coach / feedback
Scrum of Scrums / Process coach / feedback
Sprint Demo / Retrospective / Planning / Process coach / feedback
Scrum of Scrums / Process coach / feedback
Final presentation

Sprint 1 * (2)  Sprint 2 * (2)  Sprint 3 * (2)

* Daily Scrum meetings
## Summary of Project Statistics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of planned stories*</td>
<td>10</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Number of stories implemented by the Scrum team and accepted by the Product Owner</td>
<td>1</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Planned work hours*</td>
<td>59.25</td>
<td>82</td>
<td>153.5</td>
</tr>
<tr>
<td>Actual work hours done*</td>
<td>46.75</td>
<td>77.5</td>
<td>67.5</td>
</tr>
<tr>
<td>% of tasks estimated*</td>
<td>80%</td>
<td>75%</td>
<td>75%</td>
</tr>
<tr>
<td>Tasks closed / Total number of tasks*</td>
<td>36/41</td>
<td>43/63</td>
<td>17/61</td>
</tr>
<tr>
<td>Quality of planning*</td>
<td>73%</td>
<td>38%</td>
<td>70%</td>
</tr>
</tbody>
</table>

*RTC DATA

45 User Stories
Burndown Charts for Sprints 1, 2 and 3
<table>
<thead>
<tr>
<th>Practice / Principle / Artifacts</th>
<th>What Worked Well</th>
<th>What Was Problematic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Planning</td>
<td>Tool – The developers were familiar with how to set up Sprint Backlogs in RTC by Sprint 2.</td>
<td>Process – Estimates did not improve over the Sprints due to unrealistic implicit goals of the students (developer heroes). Absences were not factored into the planning of Sprint 2. Late planning in Sprints 2 and 3 due to holidays and exams.</td>
</tr>
<tr>
<td>Scrum Roles</td>
<td>Rotating Scrum Master</td>
<td>Process – Scrum Masters did not facilitate Scrum Reviews, which led to delays and absence of working software to demonstrate. Tool – No visibility as to who is the Scrum Master.</td>
</tr>
<tr>
<td>Scrum Meetings</td>
<td>Daily Scrum</td>
<td>Process – Scrum Masters were not done regularly, which reduced visibility for the Process Coach and the whole team. Reasons for impediments were not detailed enough to act upon. Inconsistencies in the chronology led to confusion. Tool – Team member absences are not automatically populated.</td>
</tr>
<tr>
<td>Sprint Demo</td>
<td>Process – The developers realized at the final Sprint demo that demonstrating software remotely requires preparation.</td>
<td>Tool – No software was demonstrated in any of the Sprint reviews. A technically savvy Product Owner had to check out the current version during Sprint demos. Videos and screenshots were not prepared.</td>
</tr>
<tr>
<td>Practice / Principle / Artifacts</td>
<td>What Worked Well</td>
<td>What Was Problematic</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td><strong>Scrum Artifacts</strong></td>
<td>Tool – The developers chose the high priority user stories to work on. Team got an organized set of wikis for Scrum artifacts (e.g., Process coach feedback, Sprint retrospective and code convention).</td>
<td>Process – Granularity of the task decomposition was too coarse so tasks stayed open for a very long time. User stories were dragged from Sprint to Sprint without looking at velocity. Tool – Product backlog and Sprint backlogs are not presented in a straightforward way. Tasks related to stories are not presented together.</td>
</tr>
<tr>
<td><strong>Communications</strong></td>
<td>Process – Students shared screen shots of the product to achieve consistency of the user interface. In Sprint 2, students managed to have a chat all together.</td>
<td>Process – No time was used for deciding how to work more smoothly together and integrate work, dragging problems from Sprint 1 to Sprint 3. The time difference between the three countries was problematic.</td>
</tr>
<tr>
<td><strong>Empower the Team</strong></td>
<td>Process – The developers decided on the stories they wanted to implement in each Sprint.</td>
<td>Process – The developers did not work as one team, but as three smaller separate teams.</td>
</tr>
</tbody>
</table>
Research Questions

- **Role of the Process** -- How well do Agile and Scrum practices support the work of distributed developers?
  - Increase transparency
  - Time to factor for students to learn
  - Time to factor for instructors to monitor the process
  - Discipline that requires training
  - Crucial in the delivery of the final product

- **Role of the Tooling** -- How important is tooling in supporting distributed developers using Agile and Scrum practices?
  - Crucial for team awareness and delivery of the final product – Not possible otherwise

- **Guidelines** -- How are Agile and Scrum practices best introduced into distributed students’ projects?
## Introducing Scrum in Students’ GSD Projects

<table>
<thead>
<tr>
<th>Planning</th>
<th>Facilitating and Monitoring</th>
<th>Reflecting</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Define a Scrum scenario – Sprint roles, artifacts and meetings &lt;br&gt; - Establish a strong relationship with and involve a professional certified Scrum Master and an external Product Owner &lt;br&gt; - Select a real project &lt;br&gt; - Identify the constraints &lt;br&gt; - Assess the risks &lt;br&gt; - Select tools &lt;br&gt; - Determine research objectives &lt;br&gt; - Set-up data collection instruments &lt;br&gt; - Prepare tutorials and evaluate students &lt;br&gt; - Train students (e.g., XP game) &lt;br&gt; - Have students sign a net etiquette form</td>
<td>- Organize a jump start meeting for the project &lt;br&gt; - Organize socialization activities involving all team members &lt;br&gt; - Facilitate Scrum meetings / Scrum retrospectives and demo reviews &lt;br&gt; - Monitor the Scrum artifacts &lt;br&gt; - Mix synchronous and asynchronous communications &lt;br&gt; - Have students be prepared for meetings &lt;br&gt; - Take notes of what is happening on the project</td>
<td>- Formally close the project with thanking the different actors involved &lt;br&gt; - Summarize what went well on the project and what didn’t, and determine how to refine the model</td>
</tr>
</tbody>
</table>
USA
Developers (9)
Product owner (3)
Auditors (21)
Process coach (instructor) (1)

Senegal Testers (11)

India Testers (2)

Cambodia Testers (2)

2010

End-to-End Tooling
Scrum
Agile Methodologies

Quality

Mobile

Quality

End-to-End Tooling
Scrum
Agile Methodologies

University of Delhi

Ecole Supérieure Multinationale des Télécommunications

Tools:
- git
- redmine
- eclipse
- Android
- RALLY
- Java
- Google Talk
- Google Groups

Methodologies:
- Scrum
- Agile
Software Development Projects

- **Project 1: No Ink** for Blackberry phones
  - Taking, annotating and organizing notes

- **Project 2: Back Pocket** for basic phones
  - Budgeting for students

- **Project 3: Godiva Flash Cards** for Android phones
  - Social flash cards to revise class topics

End-to-end tooling infrastructure
Thanks

- All students involved to date
- NCIIA
- IBM
- Pace University