FULLY DISTRIBUTED SCRUM: THE SECRET SAUCE FOR HYPERPRODUCTIVE OUTSOURCED DEVELOPMENT TEAMS


Jeff Sutherland, Ph.D.
Co-Creator of Scrum
jeff.sutherland@scruminc.com
Chairman, Scrum Training Institute

Guido Schoonheim
CTO Xebia
gschoonheim@xebia.com
About Xebia

- Holland, France, India
- 160 people
- Agile & Java focus
- What we do:
  - Software development
  - Agile training & consultancy
  - Agile offshoring

Core values:
- People first
- Customer intimacy
- Quality without compromise
- Share knowledge
Xebia OneTeam

• Since 2006, Xebia (Netherlands) started localized projects with half Dutch and half Indian team members.
• After establishing localized hyperproductivity, they move the Indian members of the team to India and show increasing velocity with fully distributed teams.
• After running XP engineering practices inside many distributed Scrum projects, Xebia has systematically productized a model similar to the SirsiDynix model for high performance, distributed, offshore teams with linear scalability and outstanding quality.
Distributed/Outsourcing Styles

Isolated Scrums

Distributed Scrum of Scrums

 Totally Integrated Scrums
Outsourcing

What happens if you outsource $2M of development?
  – Industry data show 20% cost savings on average

Outsourcing from PatientKeeper to Indian waterfall team:
  – Two years of data showed breakeven point occurs when Indian developer costs 10% of American Scrum developer
  – Actual Indian cost is 30%

$2M of Scrum development at my company costs $6M when outsourced to waterfall teams

Never outsource to waterfall teams. Only outsource to Scrum teams.
SirsiDynix - Anatomy of a failed project

Over a million lines of Java code
SirsiDynix Distributed Scrum

- 56 developers distributed across sites

- SM
  - Dev
  - Dev
  - Dev

- TLd
  - Dev
  - Dev
  - Dev

- Catalogue
- Serials
- Circulation
- Search
- Reporting

- SirsiDynix
  - Provo, Utah
  - Denver, CO
  - Waterloo, Canada

- Exigen Services
  - St. Petersburg, Russia

© Jeff Sutherland 1993-2007
Scrum daily meetings

- Local Team Meeting
- Scrum Team Meeting

St. Petersburg, Russia 17:45pm
7:45am Provo, Utah
### Velocity in Function Points/Dev month

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Months</td>
<td>54</td>
<td>540</td>
<td>827</td>
</tr>
<tr>
<td>Lines of Java</td>
<td>51,000</td>
<td>58,000</td>
<td>671,688</td>
</tr>
<tr>
<td>Function Points</td>
<td>959</td>
<td>900</td>
<td>12673</td>
</tr>
<tr>
<td>Function Points per Dev/Mon</td>
<td>17.8</td>
<td>2.0</td>
<td>15.3</td>
</tr>
</tbody>
</table>

1. M. Cohn, User Stories Applied for Agile Development. Addison-Wesley, 2004
Should we look at excellent Scrum teams?

- Often extreme data points are not sustainable.
- The most productive team ever recorded at Borland produced a failed product.
- The most productive distributed team (SirsiDynix) had quality problems, management problems, and internal company conflicts that caused the product to be killed.
- The second most productive team in the world (Motorola - David Anderson data) was overwhelmed with bureaucracy, completely demotivated, their product was killed, and the team died a painful death.
SirsDynix Challenges

- ScrumButt
- Builds were stable only at Sprint boundaries
- ScrumMasters, Product Owners, and Architects only in U.S.
- No XP in U.S, only in Russia
- No face to face meetings
- Low test coverage
- Poor refactoring practice
- Did not have equal talent across teams
- Company merger created competitive products
- Sirsi now owned Dynix and killed Dynix product
Research Issue

- SirsiDynix was a retrospective study of a single data point
- Even if quality was perfect, it does not prove anyone else can do it.
- Even worse, if you observe a finding after the fact, you cannot infer causality
- Is SirsiDynix a lucky accident? Or maybe an unlucky accident?
We needed a prospective study

• Define the distributed team model before projects start
• Assure consistent talent, tools, process, and organization across geographies
• Establish high quality data gathering techniques on velocity, quality, cost and environmental factors.
• Run a consistent team model on a series of projects and look for comparable results
• Demonstrate that local velocity = distributed velocity
• Demonstrate that local quality = distributed quality
• Demonstrate linear scaling at constant velocity per developer
Aren’t Agile and Offshoring like oil and water?

• Scrum Hyperproductive teams are collocated with shared ownership, shared responsibility, high interaction

• And then you want to distribute them halfway around the planet? Are you crazy?
Mixing both worlds makes a killer combination
Mixing both worlds makes a killer combination

- Agile hyperproductivity and quality combined with offshoring benefits:
  - Cost reduction
  - Availability of talent
  - Scaling up/down with knowledge retention and without local layoffs
Mixing both worlds makes a killer combination

- Agile hyperproductivity and quality combined with offshoring benefits:
  - Cost reduction
  - Availability of talent
  - Scaling up/down with knowledge retention and without local layoffs

- Solution: Fully Distributed Scrum
Case study: Building a new railway information system
Getting in the same mindspace

We are looking for:
• Shared ownership
• Shared context
• Personal relationships
• Team culture and standards
• Shared Agile value system

We need to tie people closely together with shared goals: SCRUM!
Forming the Fully Distributed team

• Shared ownership from the start
• Decide architecture together
• Get to know the client and domain
• Norming session for the team
• Form personal relationships

Establish local hyperproductivity
Distributed Scrum Meetings

- Video conferencing is a must!
- Same Scrum rules apply
- Planning poker over video or with digital tool
- Digital Scrum boards

The single most important thing the Scrum cycle facilitates is communication!
A day in the life...

Local standup

Stand up

Stand up

India

Update Scrum board

NL

Update Scrum board

© 2008 Xebia B.V.
Scaling the project
Scaling the project
Scaling the project
Scaling the project
Some work is hard to distribute

• Software architecture distributes easily enough
• Enterprise architecture often does not
• Xebia rule: client gets same experience as with a local team
Linear productivity increase

Velocity

- India onsite
- Distributed scaling
- Team has settled

© 2008 Xebia B.V.
Quality by Definition of Done & XP

- Facts:
  - 95% found in iteration
  - 50 in acceptance
  - 0.5 – 1.0 per kLOC
### Dutch Velocity vs. Russian Velocity

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Months</td>
<td>827</td>
<td>125</td>
</tr>
<tr>
<td>Lines of Java</td>
<td>671,688</td>
<td>100,000</td>
</tr>
<tr>
<td>Function Points</td>
<td>12673</td>
<td>1887</td>
</tr>
<tr>
<td>Function Points per Dev/Month</td>
<td>15.3</td>
<td>15.1</td>
</tr>
</tbody>
</table>

1. M. Cohn, User Stories Applied for Agile Development. Addison-Wesley, 2004
Conclusion
Conclusion

• Fully Distributed Scrum has the full benefits of both local hyperproductive teams and offshoring
Conclusion

• Fully Distributed Scrum has the full benefits of both local hyperproductive teams and offshoring

Fully Distributed Scrum has more value then localized Scrum