Critical Success Factors in Distributed Agile for Outsourced Product Development

December 2009

CONSEG-09 International Conference on Software Engineering, India
Agenda

- OPD Environment - Key Characteristics
- Premise - Agile Adoption
- Facts - Distributed Agile
- Derivatives - Critical Success Factors
## OPD Environment - Key Characteristics

<table>
<thead>
<tr>
<th>Category</th>
<th>Characteristics</th>
</tr>
</thead>
</table>
| Technology | • Current and Next Generation  
              • Multiplatform, Highly Dynamic and Evolving                                   |
| Product    | • Tight Budgets, Speed to Market, Very high COQ  
              • Unbounded Users, Unpredictable Scalability                                      |
| People     | • Deep Engineering Skills  
              • Product Engineering Mindset                                                    |
| Process    | • Home-grown / Agile or Lean  
              • Lack of Detailed Documentation                                                  |
| Culture    | • Technology Driven  
              • Aggressive and Action Packed                                                   |
Premise - Agile Adoption

Agile has been the *mantra* of success in OPD.

Adoption of Distributed Agile continues to raise.

The challenges of Distributed Agile prevail.
Distributed Agile does not mean step-by-step implementation of an agile methodology.

Collaborative collation of agile practices and processes evolves a distributed agile methodology.

While the basic tenets remain intact, the implementation details vary across ecosystems.
Derivatives - Critical Success Factors

- Lessons Learned
- Best Practices
- Thought Leadership

Critical Success Factors
A. Setup the Base Camp!

Clarity on technical environment, tools and key engineering processes propel distributed agile projects.

Rapport building assures efficiency among virtual teams throughout the lifecycle.
B. Ensure Explicit Delegation and Validate Assumptions

Be explicit. Ask questions, seek clarity and never assume.

Clarity enables virtual teams deliver in agile environments.
C. Cut Communication Loops

Resolve long running unproductive communication loops that impact productivity in distributed agile teams!
D. Facilitate Tool Driven Query Resolution

Tools enable tracking, improve effectiveness and preserve knowledge. Email is not an efficient query resolution tool for agile teams.
E. Initiate Test Drives

Test Drives ensure adequate visibility and predictability on product health at every critical milestone.

Test Drives across locations increase collaboration and ownership. Feedback comes from all locations!
F. Assess Internal Quality

Code quality improves agility in distributed teams. Very essential to avoid large refactoring!
G. Manage Effort Variance Constructively

Constructive approaches improve team morale, encourage distributed teams to collaborate and reveal areas of improvement!
H. Take Stock of User Stories for Status Checks

Track each story to completion. Keep an inventory of all stories and states (Defined, Coded, Unit Tested,...).
Follow a systematic approach for RCA. Involve stakeholders across locations. This ensures effective inputs to succeed in future iterations.
J. Complement People to Improve Process

- Appreciate on time.
- Appreciate suggestions!
- Empower engineers with new roles.
- Appreciate in front of all virtual teams!
Summary

A. Setup the Base Camp!
B. Ensure Explicit Delegation and Validate Assumptions
C. Cut Communication Loops
D. Facilitate Tool Drive Query Resolution
E. Initiate Test Drives
F. Assess Internal Quality
G. Manage Effort Variance Constructively
H. Take Stock of User Stories for Status Checks
I. Invest in Root Cause Analysis
J. Complement People to Improve Process.
Successful Customers | Innovative Solutions | Happy People

Imagination  Action  Joy