



*Thinking Distributed to  
Improve Agility*

Jamie Allsop | [www.agile-trac.org](http://www.agile-trac.org)

# Setting the Scene...

- Developing in C++ for over 15 years
- Running teams for a large portion of that
  - from multi-million dollar market data products in Stock Exchanges to specialised DSP systems with global branding
- Distributed Agile teams for over 7 years
  - First experience of real Agility
- Later worked in co-located settings and struggled with agility

# Setting the Scene...

- Developing in C++ for over 15 years
- Running teams for a large portion of that
  - from multi-million dollar market data products in Stock Exchanges to specialised DSP systems with global bandwidth
- Distributed systems
  -
- Large scale in co-located settings and struggled with agility

The Question is Why??

# Challenges Faced by Distributed Teams

- Communication burden is almost unbearable
- Clear communication is compounded by cultural and language differences
- Difficult to build trust among team members
- Hard to share and maintain tacit knowledge
- Synchronisation is often problematic
- Ensuring fair and accessible participation to all team members
- Maintaining a shared vision across all members often a challenge
- Providing visibility into progress for stakeholders both inside and outside of the organisation
- Creating and sustaining a team identity, both within the team and throughout the organisation

# Challenges of Distributed Teams not Insurmountable

- Over time these can be overcome
- Spoke about successful distributed agile teams at Agile 2008
- In fact Agile 2008 had a whole track on Distributed Agile
- However the three main takeaways were:
  - Co-location
  - Frequent face-to-face meetings
  - Always have a co-located kick-off

# Clearly Co-location is just so much simpler?

- What challenges are faced by co-located teams?
- ...
- In 2007 I started working with co-located teams
- In reality I found it harder to transition to agile and harder to be as agile
- Over time I started to see some issues but had difficulty identifying them

# Characteristics of Distributed and Co-located teams

- Communication burden overwhelming
- Sometimes large cultural differences
- Local Offices have differing values
- Feedback slower and constrained

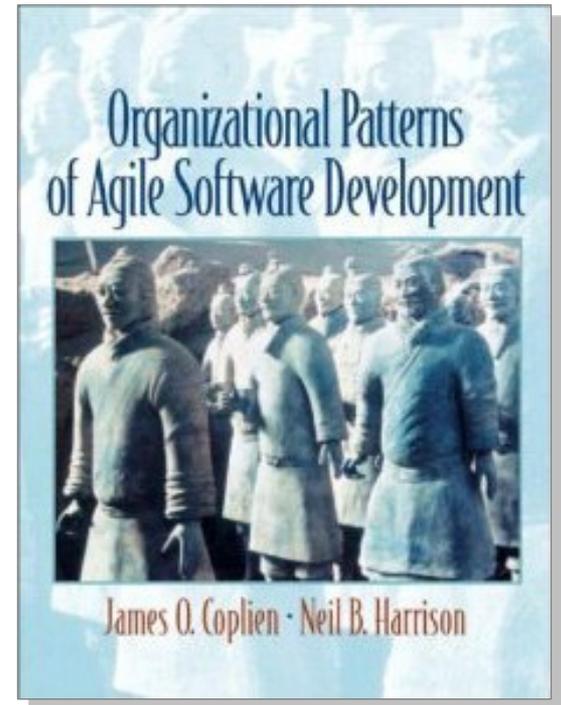
- Rich and diverse methods of communication
- Often cultural similarities
- Office shares common values
- Feedback is often rich and immediate

# Why would it be harder transitioning a co-located team to Agile?

- We need to look at organisational structure...
- Many books talk about organisational structure and communication
- In 2005 Coplien and Harrison published, “Organizational Patterns of Agile Software Development”
- Directly relevant to our domain based on many years of study of real software organisations

# Organizational Patterns

- Collates observed organizational patterns from the software industry
- Introduces organisational pattern languages
- Relates to culture, process, structure and values
- Case Studies of real companies



# Four Organisational Pattern Languages

- Project Management
  - Organisational aspects of managing projects
- Piecemeal Growth
  - How an organisation grows and develops over time
- Organisational Style
  - The general approach to the way the organisation works
- People and Code
  - The ways in which people affect code and vice-versa
- Different views of the same organisation
- Helps understand the structure of an organisation

## *Key Point*

# Process, Structure, Values

- Process ← Structure ← Values
- Values are inherent in the individuals that comprise the organisation
  - Values are observed
  - Not the same as the values an organisation professes to have!
- Changing Process alone has minimal effect
  - You must change Structure
  - What about Values?

# Changing Structure in Co-Located Setting

- Can be hard
  - Local values and structure predominate
- Tolerance for failure is much higher in the Co-Located setting
  - Not apparent that change is needed
  - Impact of change hard to understand
  - Multiple communication and feedback channels can create noise

# Contrast with a Distributed Setting

- Values of the team are typically more varied
- Team is like a virtual sub-organisation
  - Foundational values aligned with members
  - Interfaces with wider organisation discovered
- Changing structure to support new process can be more flexible
  - The team already has looser alignment with the predominant local values
  - Team already forced to cope with differences

# Contrast with a Distributed Setting

- Values of the team are typically more varied
- Team is like a virtual sub-organisation
  - ... values aligned with members
  - ... discovered
- Changes can be more flexible
  - The team already has looser alignment with the predominant local values
  - Team already forced to cope with differences

*Possibly a better starting point?*

# *Key Point*

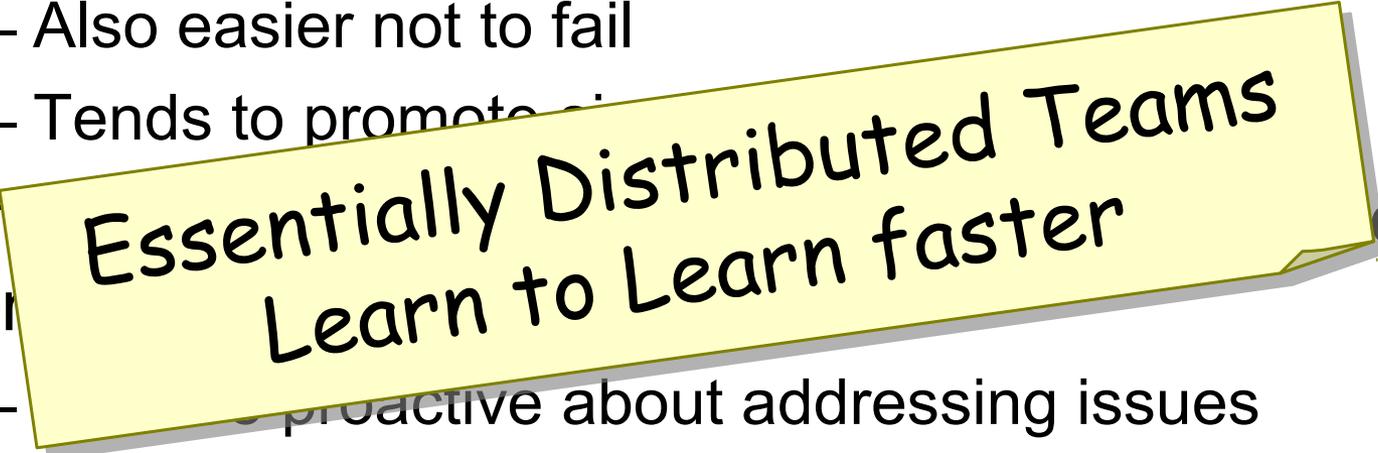
## The Ability to Learn

- Single-loop
  - Changing the *How* but not the *Why*
  - Doing the same thing better
- Double-loop
  - Focus on *Why*: Insightful Learning
  - Increase Knowledge and Understanding
- Triple-loop
  - Understand the organisation's identity
  - Learn how to evolve the organisation

# Being forced to learn better

- In a Co-Located setting it is easy to be reactive and see some improvement
  - Also easier not to fail
  - Tends to promote single-loop learning
- In the Distributed setting being reactive is not enough
  - Must be proactive about addressing issues
  - Failure comes quicker
  - Deeper understanding required
  - Promotes second-loop learning

# Being forced to learn better

- In a Co-Located setting it is easy to be reactive and see some improvement
  - Also easier not to fail
  - Tends to promote...
- In  Essentially Distributed Teams Learn to Learn faster
  - ...proactive about addressing issues
  - Failure comes quicker
  - Deeper understanding required
  - Promotes second-loop learning

## *Key Point*

# Organisational Patterns & Culture

- Each of the systemic patterns presented are part of a pattern language, the bigger picture
- It is this bigger picture and the interactions of individual patterns that constitutes the organisation's culture
- Difficulty changing structure makes it hard to evolve culture
- If you are *in* a culture it is hard to see the culture
- Makes it easy to miss important cultural roots of failures, again and again

# Promoting Culture Good or Bad?

- In the Co-Located setting a strong culture can in fact be a barrier to improvement
  - Hiring to reinforce a perceived good culture can be counter-productive
  - Often see mistakes repeated
  - But it is hard to know any better
- In the distributed setting we are more likely to accept differences and perhaps benefit from them
  - The team culture is likely evolving based on team composition

# Promoting Culture Good or Bad?

- In the Co-Located setting a strong culture can in fact be a barrier to improvement
  - Trying to reinforce a perceived good culture can be

*Cultural Influences are diluted over distance*

- In the distributed setting we are more likely to accept differences and perhaps benefit from them
  - The team culture is likely evolving based on team composition

# Sounds like Coupling and Interfaces

- Organisations and their teams are like systems and components
- The Co-located team is tightly coupled
  - Both Internally and Externally
  - Through many fat interfaces
- The Distributed team, by necessity is loosely coupled
  - Both Internally and Externally
  - Through thinner, better defined interfaces
  - To succeed these interfaces must be understood



# Sounds like Coupling and Interfaces

- Organisations and their teams are like systems and components
- The *Step* team is tightly coupled
  - *Step* is a reasonable step
- The *Step* norm to cope
  - Both Internally and Externally
  - Through thinner, better defined interfaces
  - To succeed these interfaces must be understood

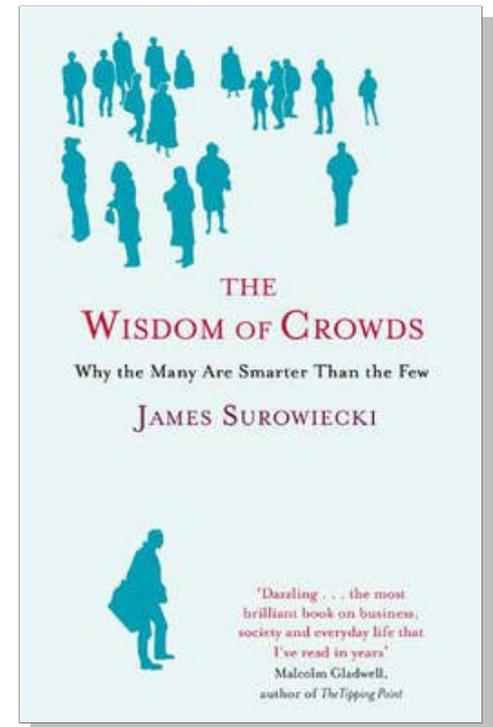
*In that context transitioning to Agile is a reasonable step*

# Why might agility be harder in a Co-Located team?

- We need to understand a little bit more about how groups work together to make good decisions
- In 2004 Surowiecki published, “The Wisdom of Crowds”
- Not a book about software development
  - about how groups of people make decisions
  - and whether those decisions are good ones
- Surowiecki gave the Keynote at Agile 2008

# The Wisdom of Crowds

- “Why the many are smarter than the few”
- Premise: How a group (crowd) of people can, *under certain conditions*, reach a smarter decision than the smartest person in the group
- Consider the weight of an Ox...
- LOC in Visual Studio...
- What about dumb crowds?



## *Key Point*

# The Elements of a Wise Crowd

- Diversity of Opinion
  - People bring different information
- Independence of Opinion
  - Most weight and trust given to own information
- Decentralisation of Knowledge
  - No-one dictates the expected outcome
- Meaningful Aggregation
  - So that a collective verdict can be reached

# Factors which make good group decisions hard

- All people have the same information and experience
- Centralised control over information flow
- Inability to pull needed information from information sources
- Imitation and information cascades
- Emotional reactions to outside pressures can result in herding

# Factors which make good group decisions hard

- All people have the same information and experience
- Central control over information flow
- Inability to share information
- Imitation and information cascades
- Emotional reactions to outside pressures can result in herding

*Co-located teams are at a disadvantage*

# Some Implications...

- The best decisions are achieved through competition and disagreement
  - Groups of similar people find it easier to make poor decisions
- In an agile environment we value shared vision and tacit knowledge
  - Sometimes difficult to have a collective view while maintaining independence

# Co-located Teams can Struggle

- Shared values and structure result in self-reinforcing behaviour
  - The 'collective wisdom' is often not wise
- Issues are difficult to see
- Peer pressure a real problem
- Lack of Diversity in the team
  - Easy to have ineffectual and inefficient communication
- Easier to iterate without learning
  - Practice makes permanent, not perfect

# Distributed Teams can do better

- Varied Values and Structure force looser coupling
- By definition the distributed team is outside the norm
- Greater team diversity offers the chance for greater independence
- Defined interfaces minimise noise and highlight failures faster
- Being reactive is not enough, understanding why things happen is required
- Diversity is inherent in the team

## *Key Point*

# Benefits of a Wise Crowd

- Speed of Collaborative Discovery
  - Under the right circumstances it is possible to reach better answers much faster
- Effective Coordination
  - Self-organising coordination can be optimal, out-performing deliberate attempts
- Cooperation through choice
  - Potential for much more productive relationships

## *Key Point*

# Benefits of a Wise Crowd

- Speed of Collaborative Discovery
  - Under the right circumstances it is possible to reach better answers
- **It looks like there are potential advantages to Distributed Teams**
  - ... can be optimal, performing deliberate attempts
- Cooperation through choice
  - Potential for much more productive relationships

# Learn to see beyond simple processes and practices

- Not saying that the solution to problematic co-located teams is a distributed team
- However we can learn valuable lessons from the distributed setting that can be usefully applied
- Encouraging characteristics of distributed teams in the co-located setting can really help

# Case Study – Setting the scene

- MDD – The NYSE's replacement Market Data Distribution System
  - Vastly over budget, over 6 months late
  - Co-located team engulfed in a strong local culture
  - Conventional wisdom was they needed to do more of the same, only better and faster
  - Quality was unclear, regressions a constant problem
- Drew a line in the sand and started over with a changed team
  - Introduced 3 developers working remotely in the US
  - Varying experiences and backgrounds
  - Encouraged working from home, moved to a more distributed model

# How did things change?

- We started to see problems with how we were doing development right away
  - Our discipline was poor but we couldn't get away with that with the distributed team – we had to improve
  - The new team members were able to see problems the existing team had overlooked, or simply tolerated
- The whole team was brought together to re-architect the product
- More effort was put into communication, both how we communicated and when
  - Daily stand-ups became a true heartbeat, not an irritation
  - Changing code *co-pilots* fostered closer relationships and also loosened the culture grip
  - Changes to improve intra-team visibility encouraged better code management practices

# What was the result?

- The first phase 2 release was produced after only a couple of months
  - Despite a complete re-architecture
- The first release of the newer Phase 2 system went into production over two years ago
  - Since then there has been only one production incident attributable to a bug in the system
  - It has the best reliability record of any market data system in the NYSE
  - It currently far exceeds initial performance expectations

# What was the result?

- The first phase 2 release was produced after only a couple of months

*The Distributed team, making use of diversity and knowledge out-performed the original Co-located team*

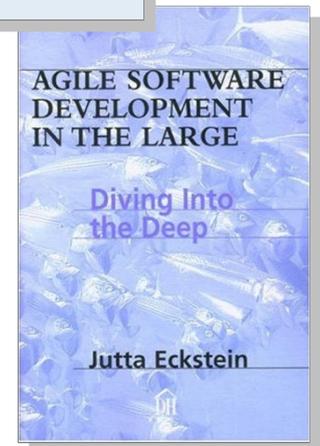
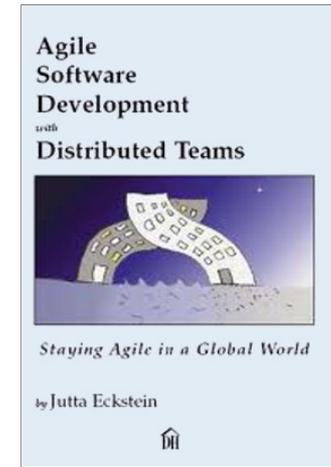
- It has the most advanced architecture
- It has the most advanced data system in the NYSE
- It currently far exceeds initial performance expectations

# Some Conclusions

- Embrace diversity in teams and consider introducing it when absent
- Ensure that local culture is not an impediment to change and improvement
- Make an effort to see failures and understand why they occur
- Understand the interfaces that exist within the team
- Understand the interfaces that exist with the rest of the organisation
- Question what they are and why they exist
- Appreciate there are benefits to both co-location and distribution
- The emphasis is usually on emulating co-location at a distance but co-located teams can also learn from distributed teams

# Further Reading & Questions

- **Organizational Patterns of Agile Software Development** by James O.Coplien and Neil B. Harrison
- **The Wisdom of Crowds** by James Surowieki
- **Agile Software Development with Distributed Teams** by Jutta Eckstein
- **Agile Software Development in the Large** by Jutta Eckstein
- <http://www.agile-trac.org>



[ja11sop@agile-trac.org](mailto:ja11sop@agile-trac.org)