Agile Distributed Development
done right

Guido Schoonheim

CTO, Xebia

gschoonheim@xebia.com

Jeff Sutherland

Co-creator of Scrum

jeff.sutherland@scruminc.com

About Xebia

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

United Kingdom
Danmark
Denmark
Nederland
België
Belgique
Deutschland
Germany
France
Italia

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.*

Outsourcing the traditional way

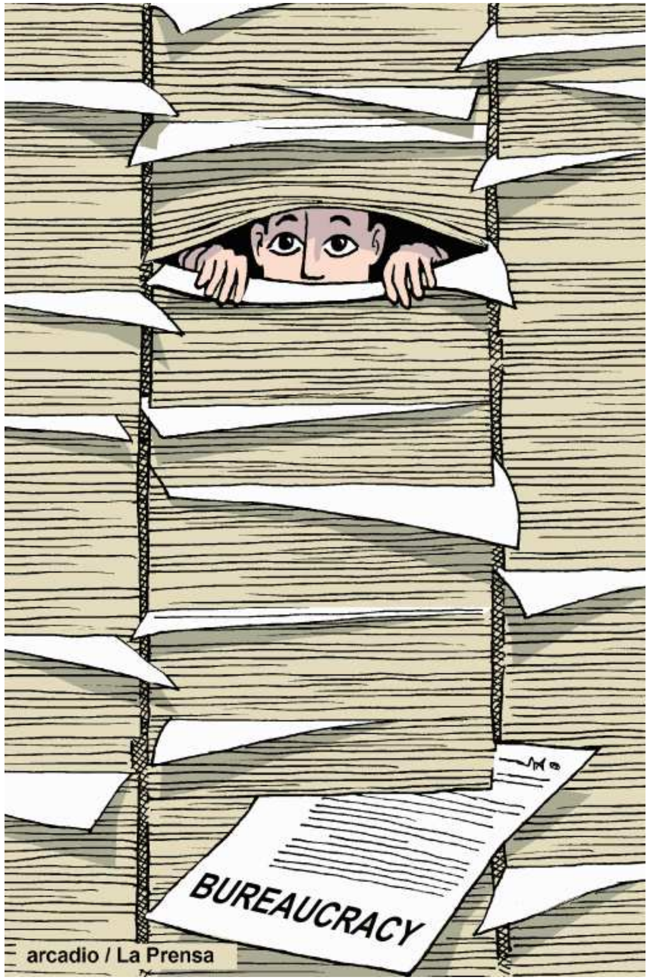
- What happens if you outsource $2M of development?
  - Industry data show 20% cost savings on average
- Outsourcing from PatientKeeper to Indian waterfall team:
  - Two years of data showed breakeven point occurs when Indian developer costs 10% of American Scrum developer
  - Actual Indian cost is 30%
- $2M of Scrum development at my company costs $6M when outsourced to waterfall teams
- Never outsource to waterfall teams. Only outsource to Scrum teams.

The real problem with distribution is indeed distance

- Geography
- Time zones
- Culture
- Language
- Different standards
- No shared ownership
- Us – Them

It’s all a form of distance ***between people***

How do we bridge that gap?



=

Or… we can look at the people

Manifesto for Agile Software Development

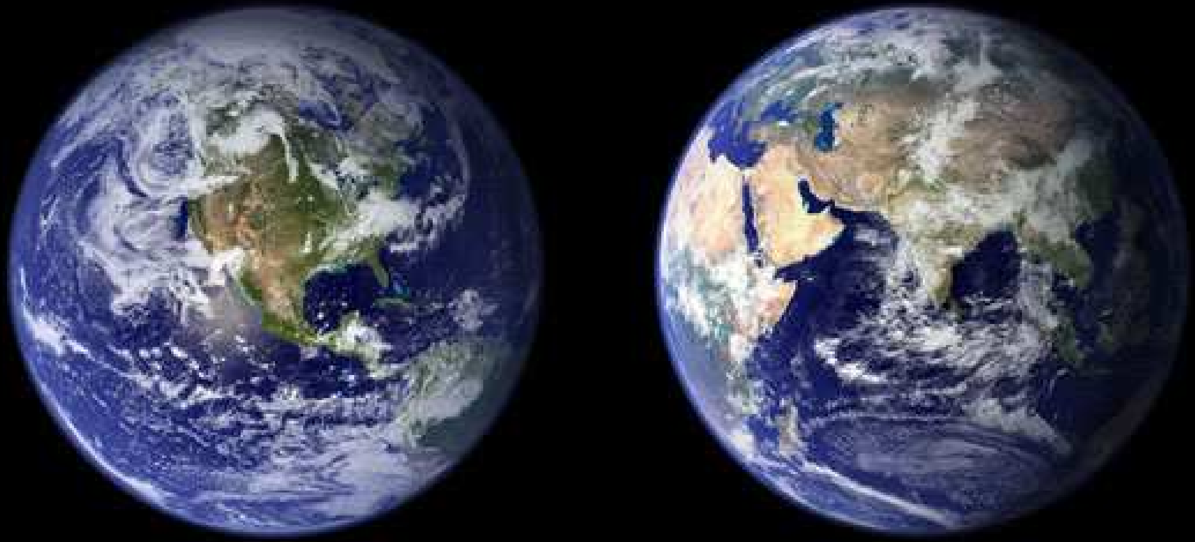
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Aren’t Agile and Offshoring like oil and water?

- Good hyperproductive Agile teams are colocated 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

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

Solution: Fully Distributed Scrum

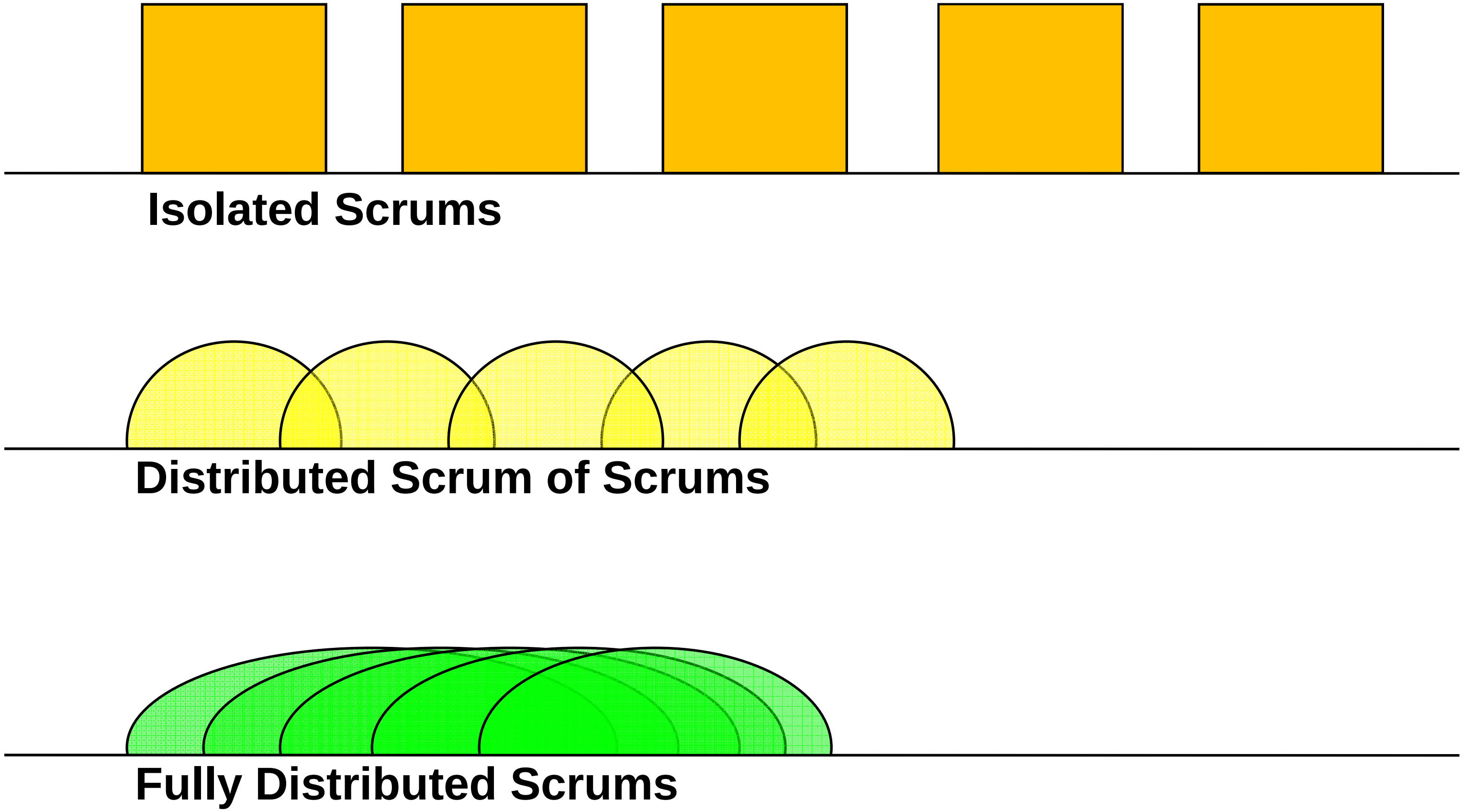
Getting in the same mindspace

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!

Distributed/Outsourcing Styles



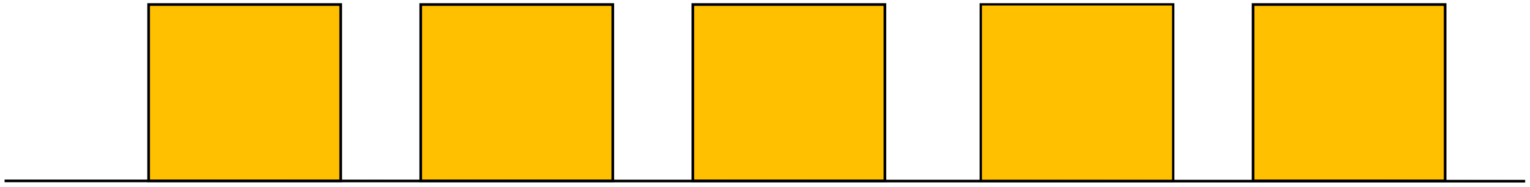
Isolated Scrums

Distributed Scrum of Scrums

Fully Distributed Scrums

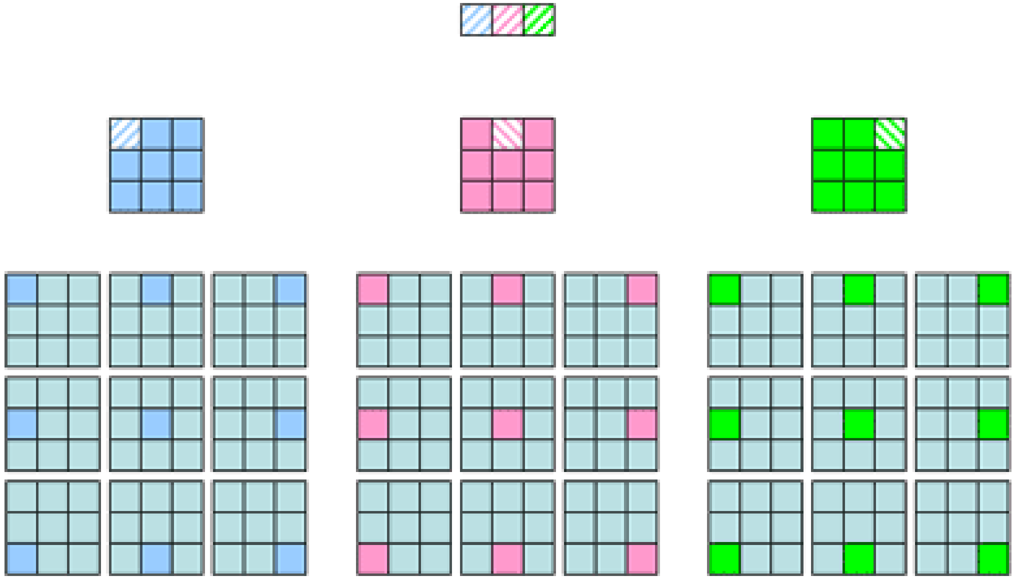
Isolated Scrums

- Works when you have local PO (=separate project)
- Otherwise you are doing waterfall with an iterative development team
- Just beats around the bush, results in classic integration problems and code ownership
- Still large degree of us / them



Scrum of Scrums

- Recommended by Scrum Alliance
- Recognizes that there is a link
- Does not give us rich communication to close the gap
- Still accepts the problem of distance as a given!



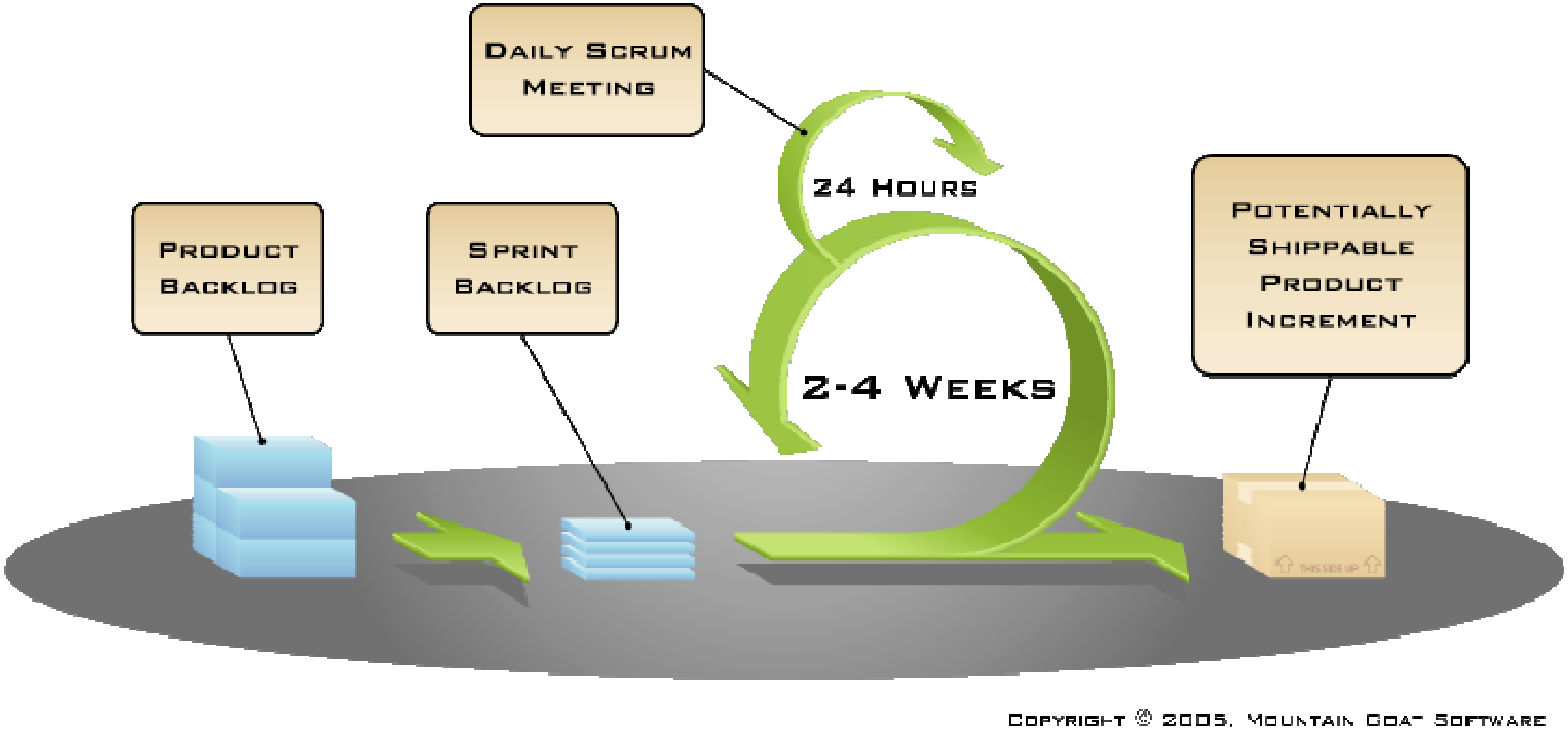
Getting in the same mindspace

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!

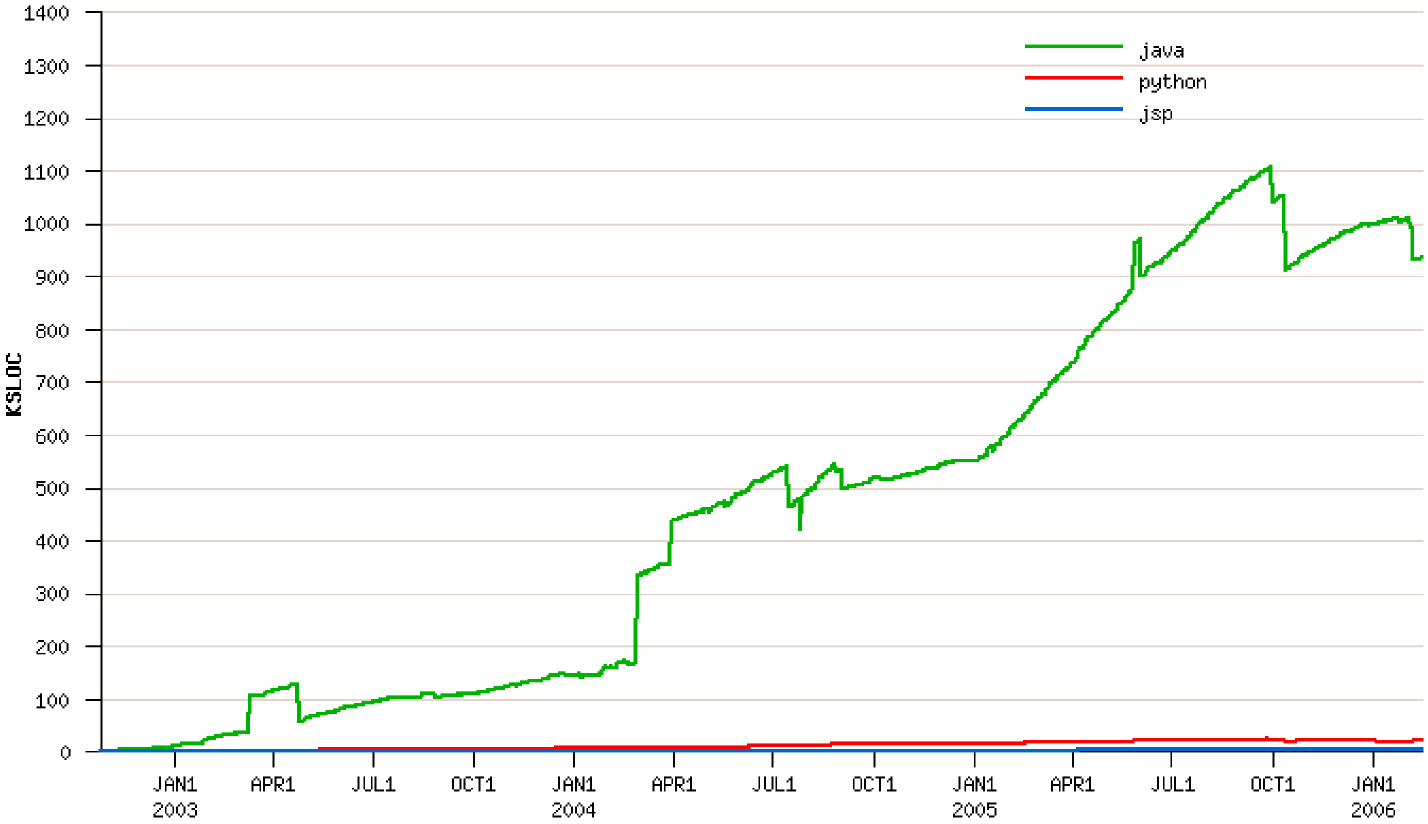
Fully Distributed Scrum


Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment
This side up
Copyright © 2005, Mountain Goat Software

Solution: Distributed Agile team members

SirsiDynix - Anatomy of a failed project

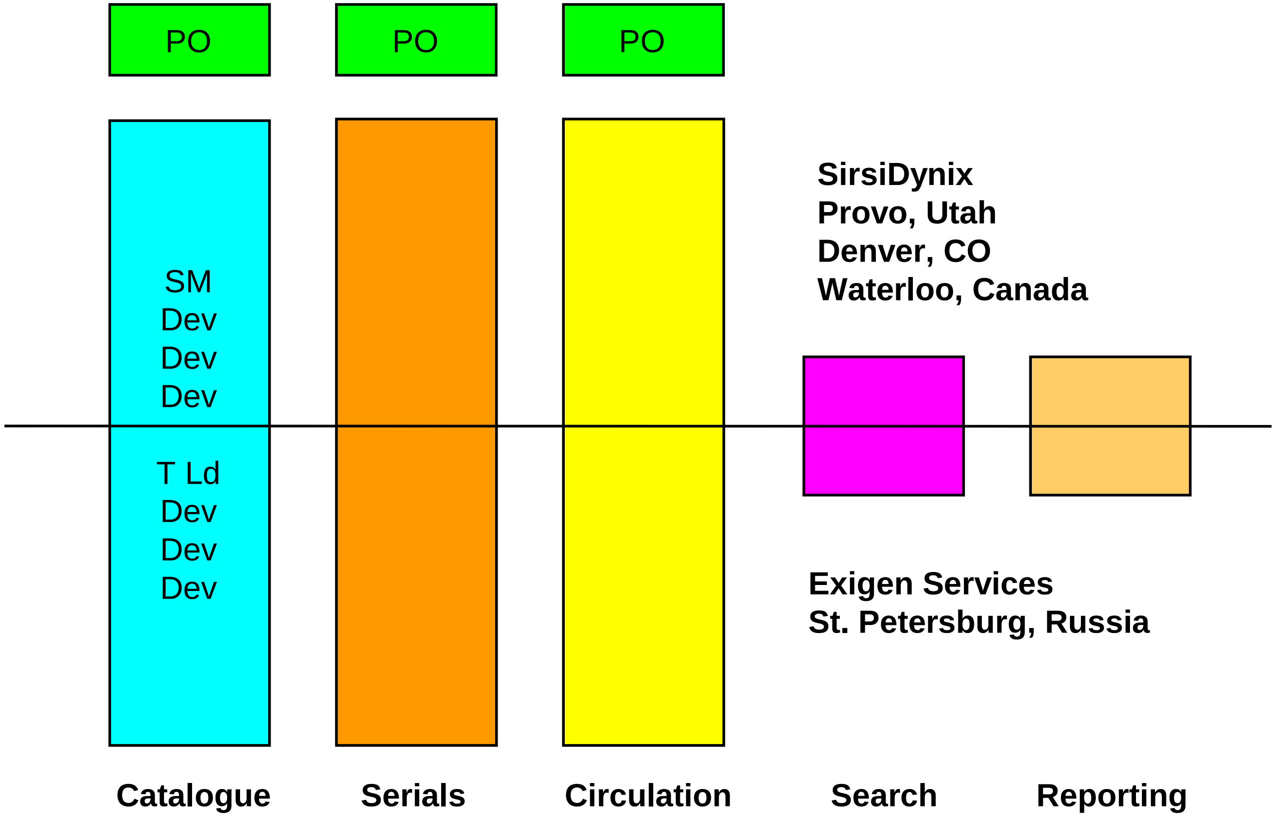
- Over a million lines of Java code



© Jeff Sutherland 1993-2007

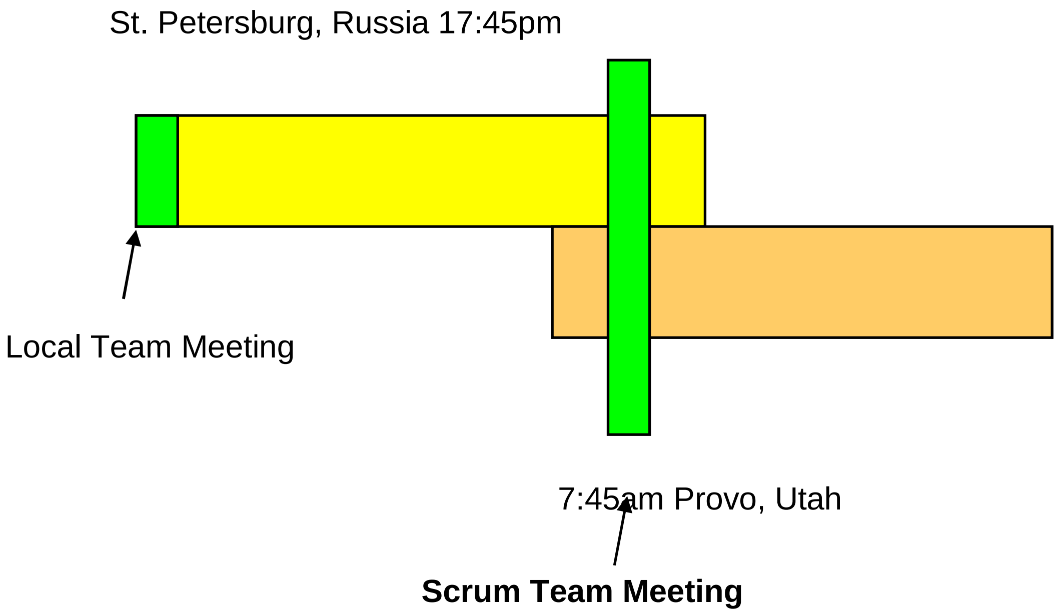
SirsiDynix Distributed Scrum

- 56 developers distributed across sites



SirsiDynix Distributed Scrum

- Scrum daily meetings



SirsiDynix 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

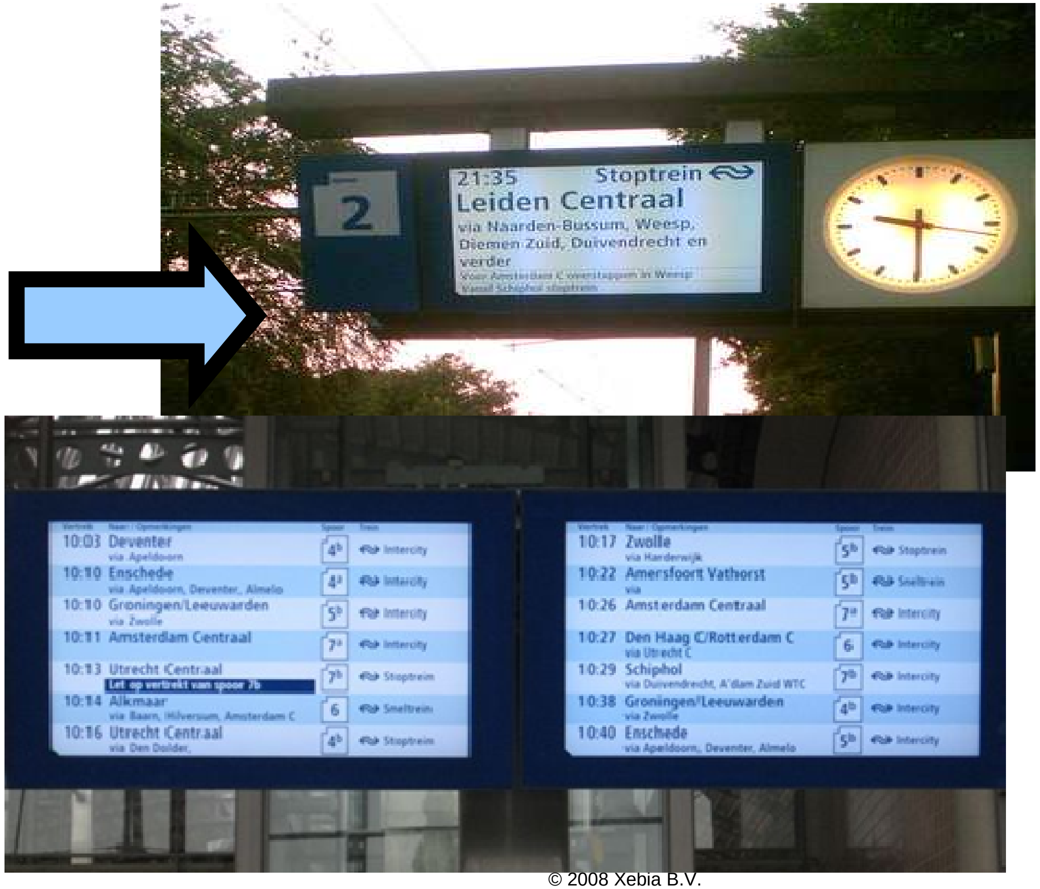
Research Issue

- 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

Case study: Building a new railway information system


2
21:35 Stoptrein
Leiden Centraal
via Naarden-Bussum, Weesp, Diemen Zuid, Duivendrecht en verder
Voor Amsterdam C overstappen in Weesp
Vanaf Schiphol stoptrein
10:03 Deventer via Apeldoorn 4b Intercity
10:10 Enschede via Apeldoorn, Deventer, Almelo 4a Intercity
10:10 Groningen/Leeuwarden via Zwolle 5b Intercity
10:11 Amsterdam Centraal 7a Intercity
10:13 Utrecht Centraal 7b Stoptrein
Let op vertrekt van spoor 7b
10:14 Alkmaar via Baarn, Hilversum, Amsterdam C 6 Sneltrein
10:16 Utrecht Centraal via Den Dolder 4b Stoptrein
10:17 Zwolle via Harderwijk 5b Stoptrein
10:22 Amersfoort Vathorst 5b Sneltrein
10:26 Amsterdam Centraal 7a Intercity
10:27 Den Haag C/Rotterdam C via Utrecht C 6 Intercity
10:29 Schiphol via Duivendrecht, A'dam Zuid WTC 7b Intercity
10:38 Groningen/Leeuwarden via Zwolle 4b Intercity
10:40 Enschede via Apeldoorn, Deventer, Almelo 5b Intercity

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

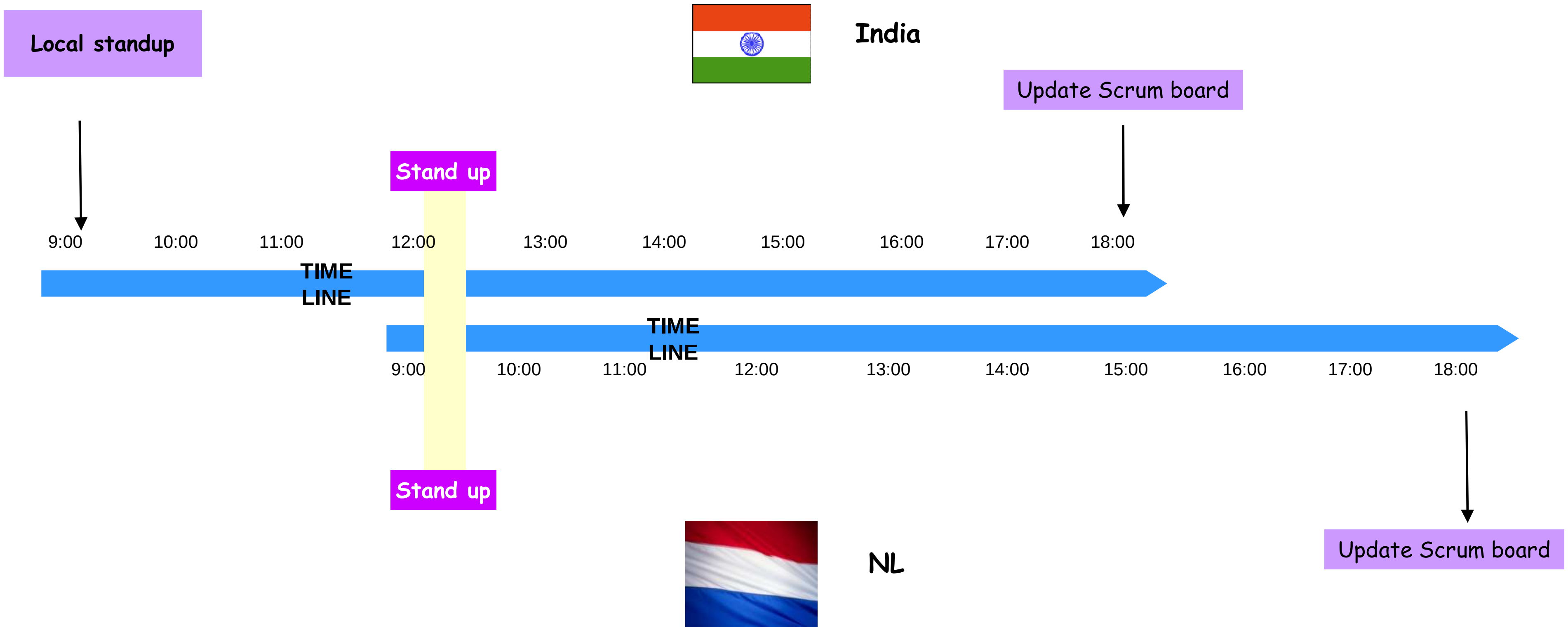
Establish local hyperproductivity

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

India

Update Scrum board

Stand up

9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00

TIME LINE

TIME LINE

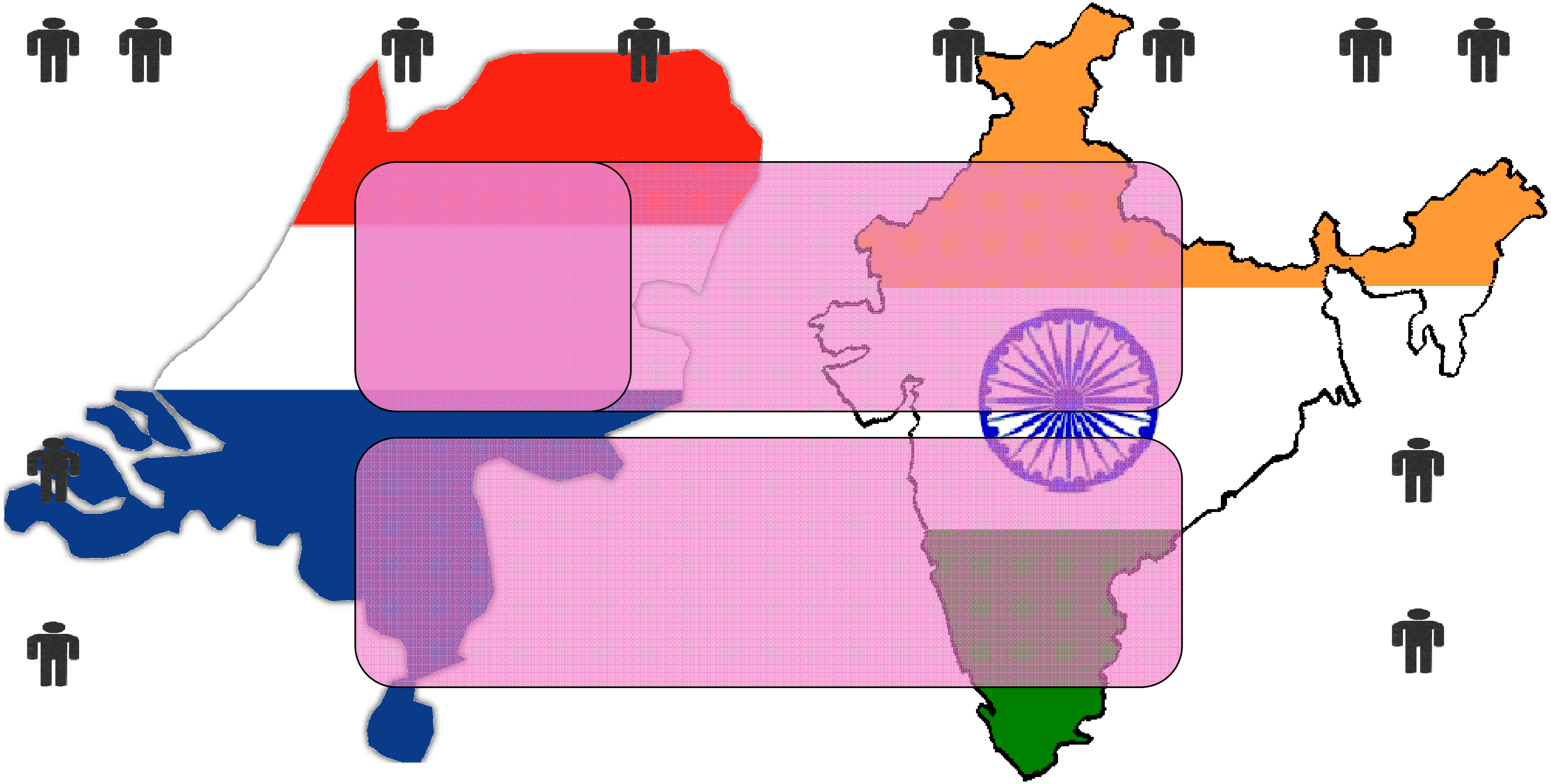
9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00

Stand up

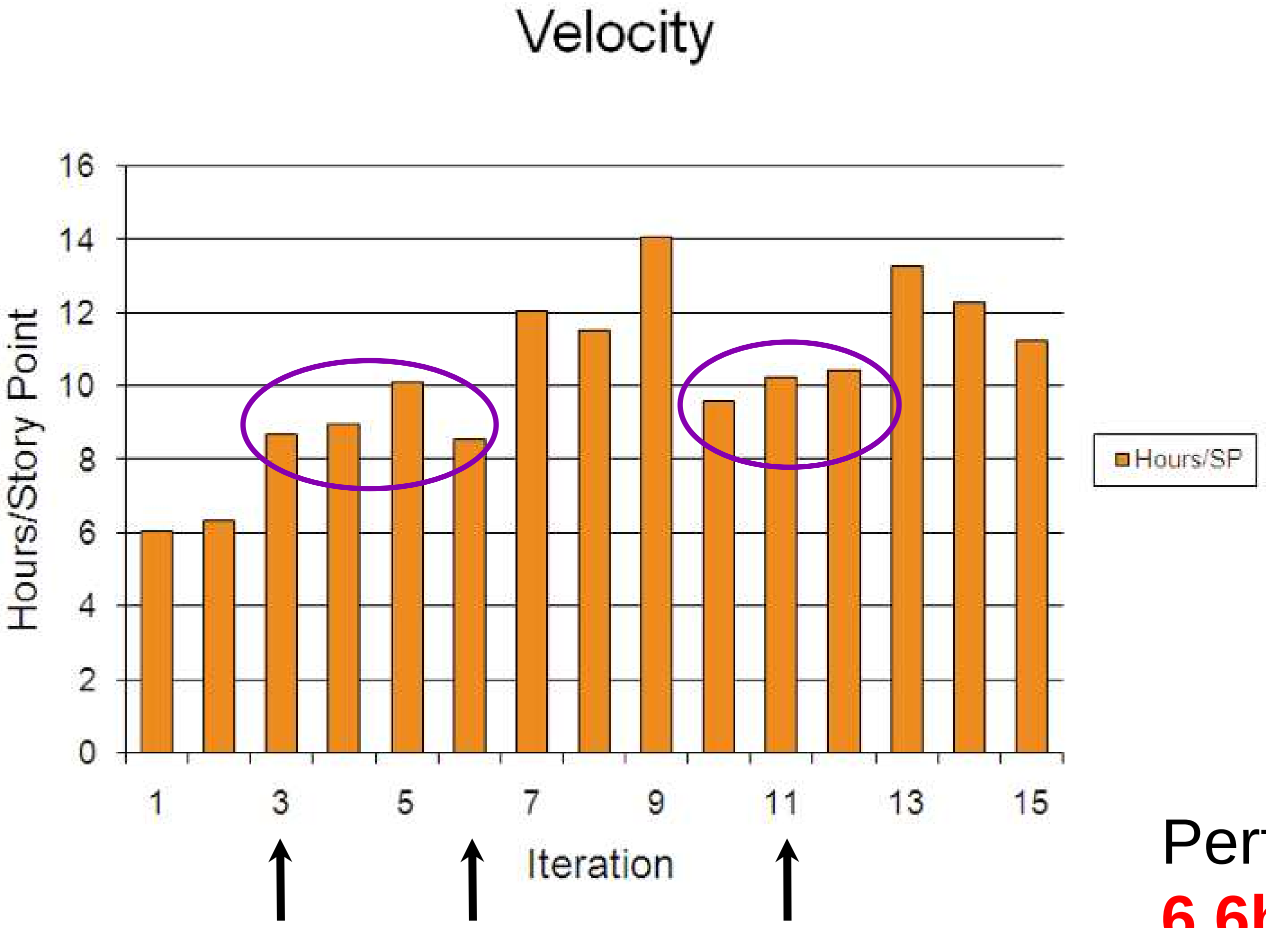
NL

Update Scrum board

Starting up a distributed team



Linear productivity increase



India onsite

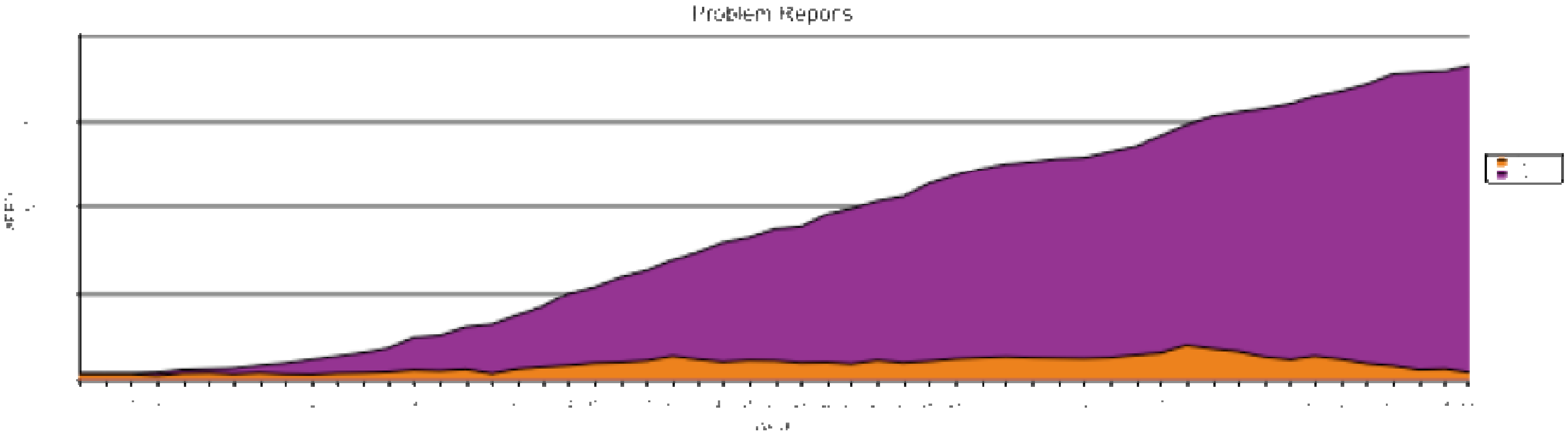
Distributed scaling

Team has settled

Performance:
6.6h/FP

Quality by Definition of Done & XP

- Facts:
  - 95 % found in iteration
  - 50 in acceptance
  - 0.5 – 1.0 per kLOC



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

Difficulties faced

- Initial reluctance to communicate extra
- Culture making it hard to get aligned, misunderstandings about priority and value
- Local team taking aggressive ownership
- Not enough context information offsite
- Keeping urgency and kaizen after adoption

Both sides need to adjust

When to start with distribution?

- Get your local organization capable of running Scrum
- Get quality up with XP practices
- Stop thrashing, focus people
- Then think of scaling up with offshore

Introducing Agile and distribution at the same time is often too much to take in! Get help!

Summary

- Working successfully in a distributed way is all about handling the ‘distance’ between **people**
- Classical approach is with more detailed instructions and control, not suited for knowledge workers!
- Agile can tie people together across distances
- Agile benefits (Time to market, performance, quality) mixed with offshoring benefits is a killer combo

Fully Distributed Scrum gives you velocity and quality **of a local team** with linear scalability

Conclusion

Fully Distributed Scrum has more value then localized Scrum

Questions?

gschoonheim@xebia.com