PRETTY GOOD SCRUM:
SECRET SAUCE FOR DISTRIBUTED TEAMS

With help from Google, Yahoo, Microsoft, IBM, Oracle, MySpace, Adobe, GE, Siemens, BellSouth, GSI Commerce, Ulticom, Palm, St. Jude Medical, DigiChart, RosettaStone, Healthwise, Sony/Ericson, Accenture, Trifork, Systematic Software Engineering, Exigen Services, SirsiDynix, Softhouse, Philips Medical, Barclays Global Investors, Constant Contact, Wellogic, Inova Solutions, Medco, Saxo Bank, Xebia, Insight.com, SolutionsIQ, Crisp, Johns Hopkins APL, Motley Fool, Planon, OpenView Venture Partners, Jyske Bank, BEC, Camp Scrum, DotWay AB, Ultimate Software, Scrum Training Institute, AtTask, Intronis, Version One, OpenView Labs, Central Desktop, Open-E, Zmags, eEye, Reality Digital, DST, Booz Allen

Jeff Sutherland, Ph.D.
Co-Creator of Scrum
+1 617 606 3652
jeff.sutherland@scruminc.com

CEO, scruminc
powered by OpenView Labs

Chairman, Scrum Training Instute

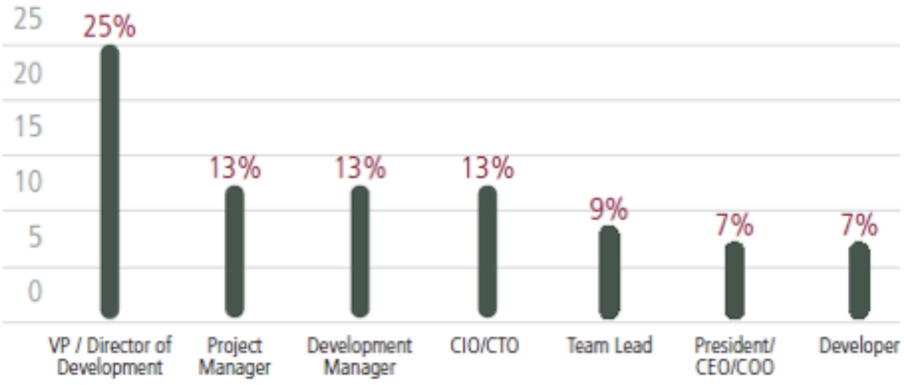
SCRUM TRAINING INSTITUTE

openview venture partners

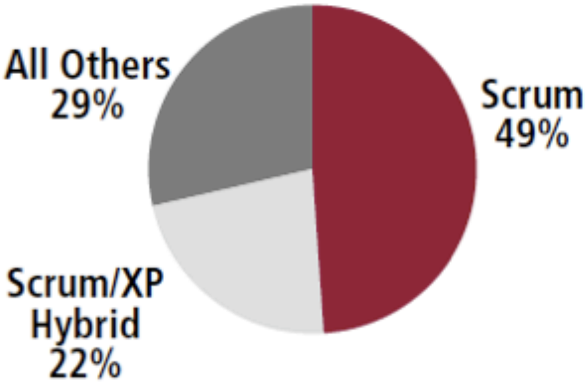
What changed in 2008?

- Scrum has become the dominant force in the Agile community
- A venture group has fully adopted Scrum for internal use and for use in all portfolio companies
  - Extreme focus on business value
  - Deeper focus on human values
  - Measuring, testing, assessing performance of teams
  - Always looking at secret sauce for hyperperformance
- Goal
  - Every aspect of the business does Scrum
  - Results change the world of venture investment
- Disruption of global business community by abandoning trust, transparency, and truth will generate extreme focus on real value.

What role most closely identifies the initial champion of Agile development within your organization?



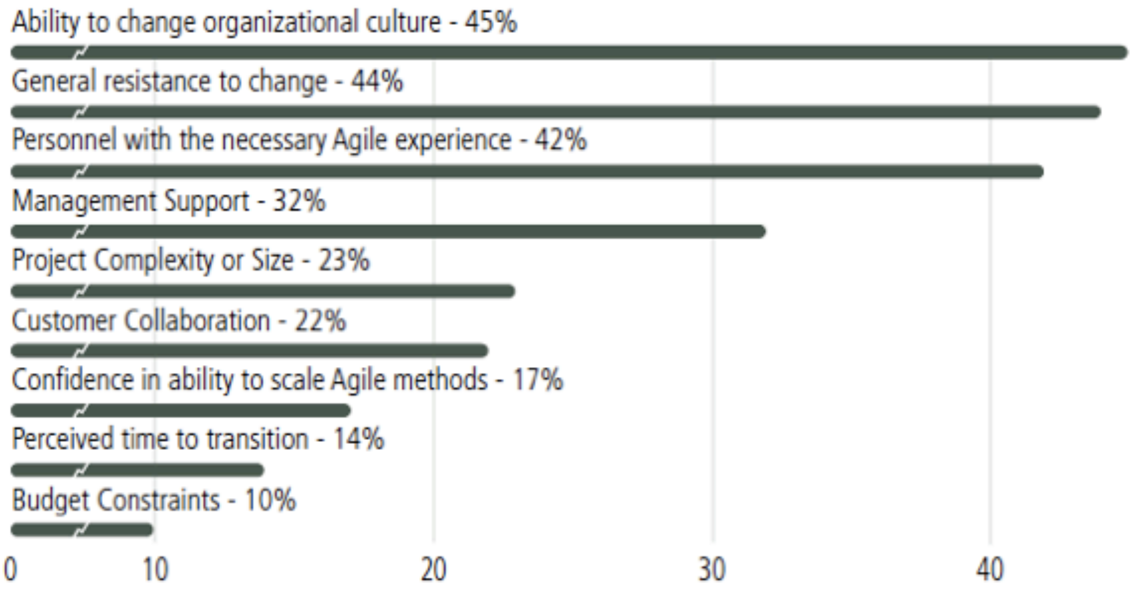
Which Agile methodology do you follow most closely?



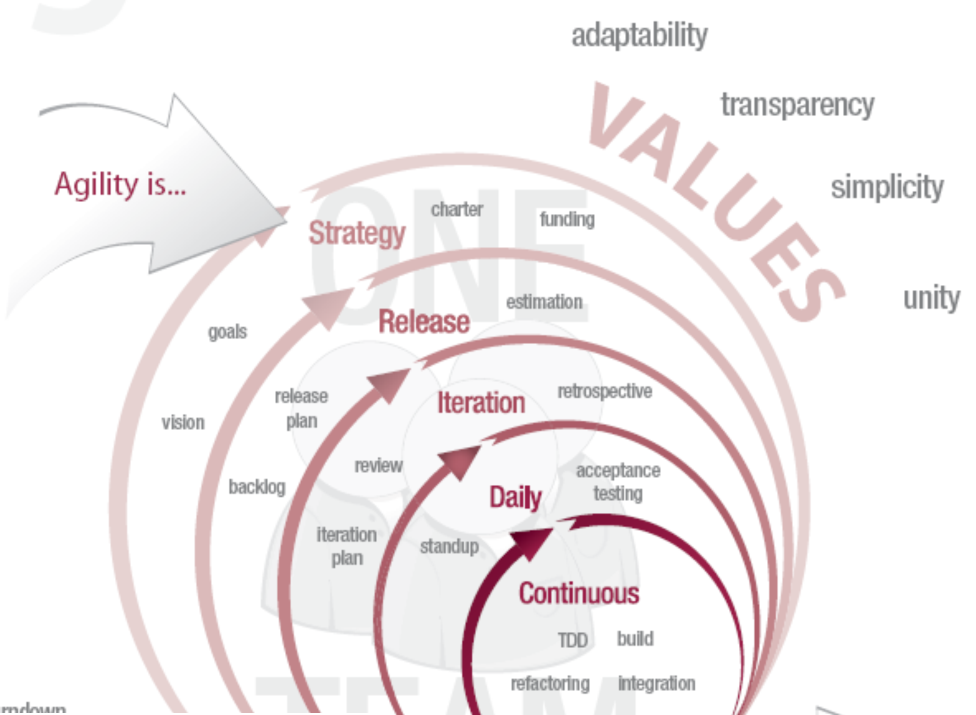
Scrum	49.1%
Scrum/XP Hybrid	22.3%
Extreme Programming (XP)	8.0%
Custom/Hybrid	5.3%
Don't Know	3.7%
Agile Unified Process (AgileUP)	2.2%
Other	2.2%
Feature-Driven Development (FDD)	2.1%
Lean Development	1.9%
Dynamic Systems Development Method (DSDM)	1.4%
OpenUP	0.6%
Agile Modeling	0.6%
Crystal	0.5%

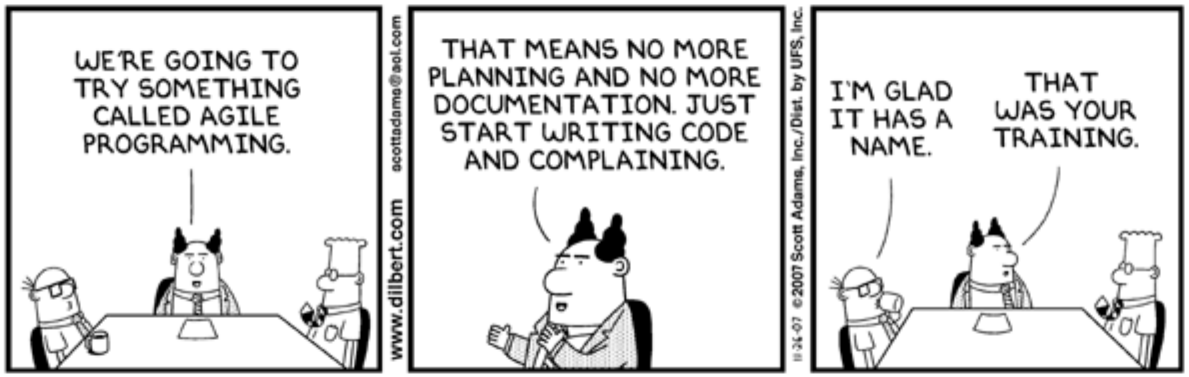
What are the barriers to further adoption of Agile in your current organization?

(select all that apply)



ScrumButt





WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
www.dilbert.com
scottadams@aol.com
11-26-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.

© Scott Adams, Inc./Dist. by UFS, Inc.

One way to measure ScrumButt

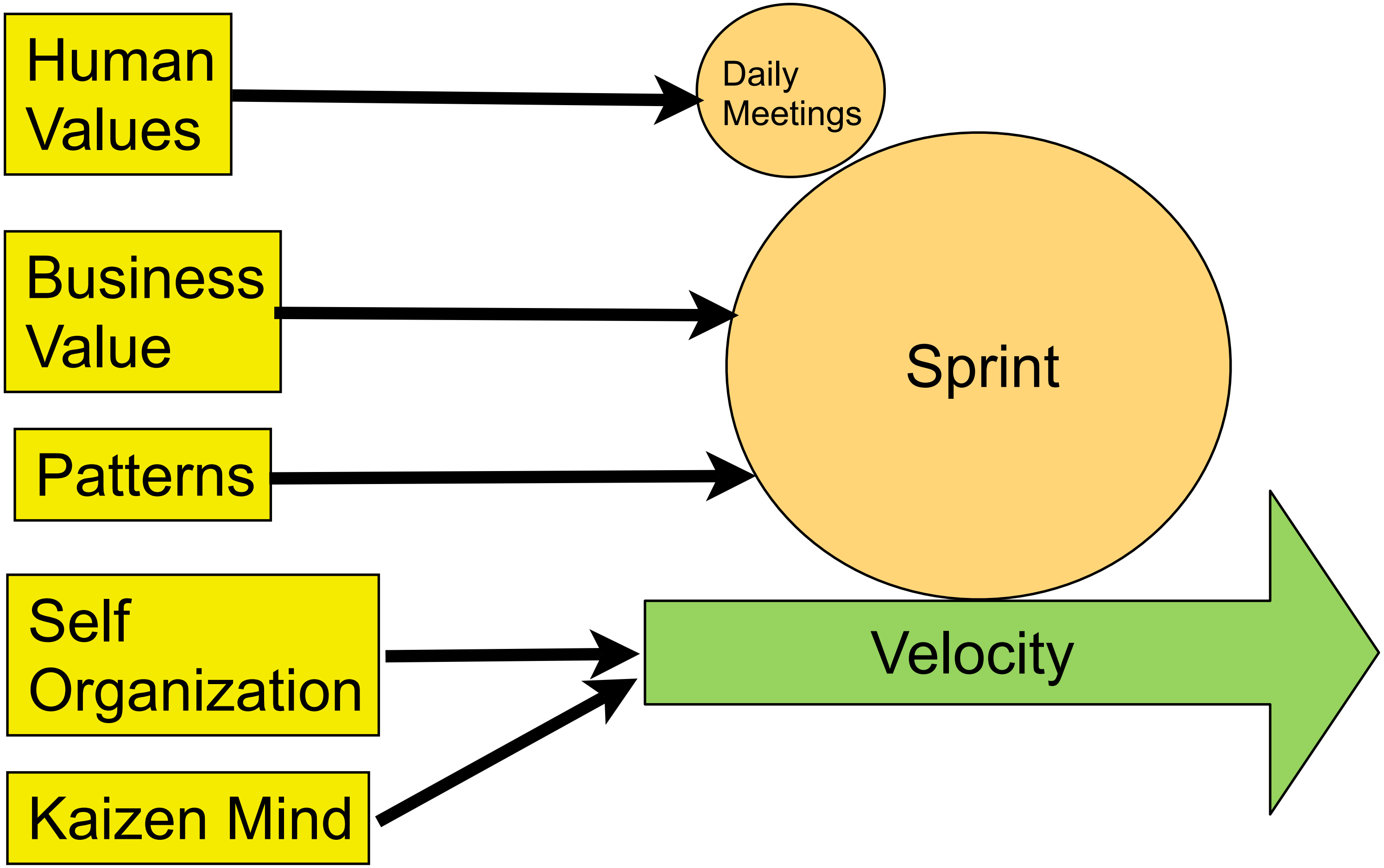
- Excellent Scrum - annual revenue up 400%
  - PatientKeeper
  - Others companies I work with in Scandinavia
- Good Scrum - revenue up 300%
  - Companies in Scandinavia I can't talk about
- Pretty Good Scrum - revenue up 150% - 200%
  - Systematic Software Engineering - 200%
  - Google - 160%
- ScrumButt - revenue up 0-35%
  - Yahoo, most companies

Venture Capital Strategy:
Follow the money

- Invest only in Agile projects
  - one hyperproductive company out of 10 might meet investment goals for a venture group
  - two or more hyperproductive could alter the market
- Invest only in market leading, industry standard processes – this means Scrum and XP
- Ensure teams implement basic Scrum practices
  - Everyone passes the ScrumButt test
  - Management held accountable at Board level for removing impediments
  - Training in secret sauce for hyperproductive teams

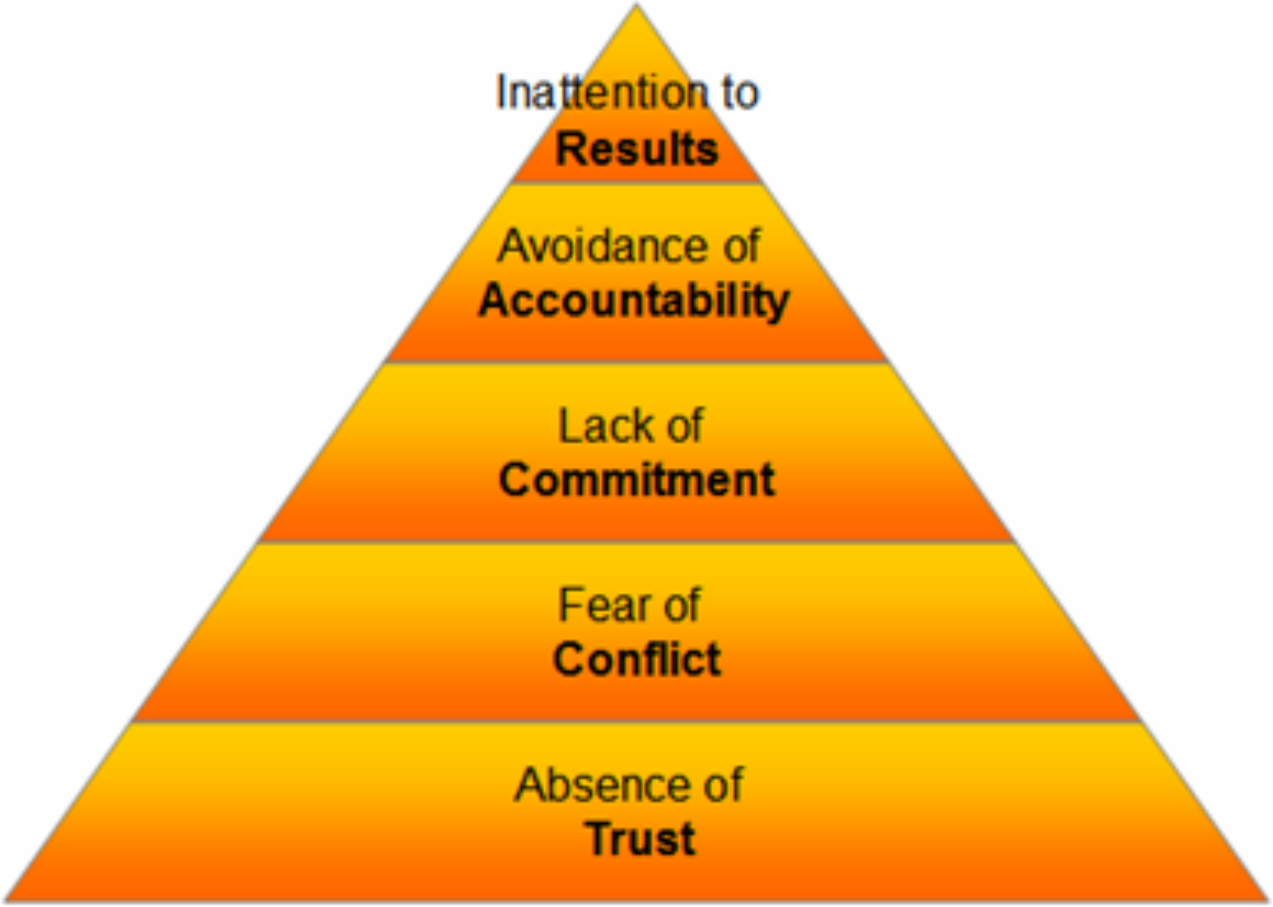
openview venture partners

Scrum Drivers



© Jeff Sutherland 1993-2008

Human Values: Five Dysfunctions of a Team



Inattention to **Results**

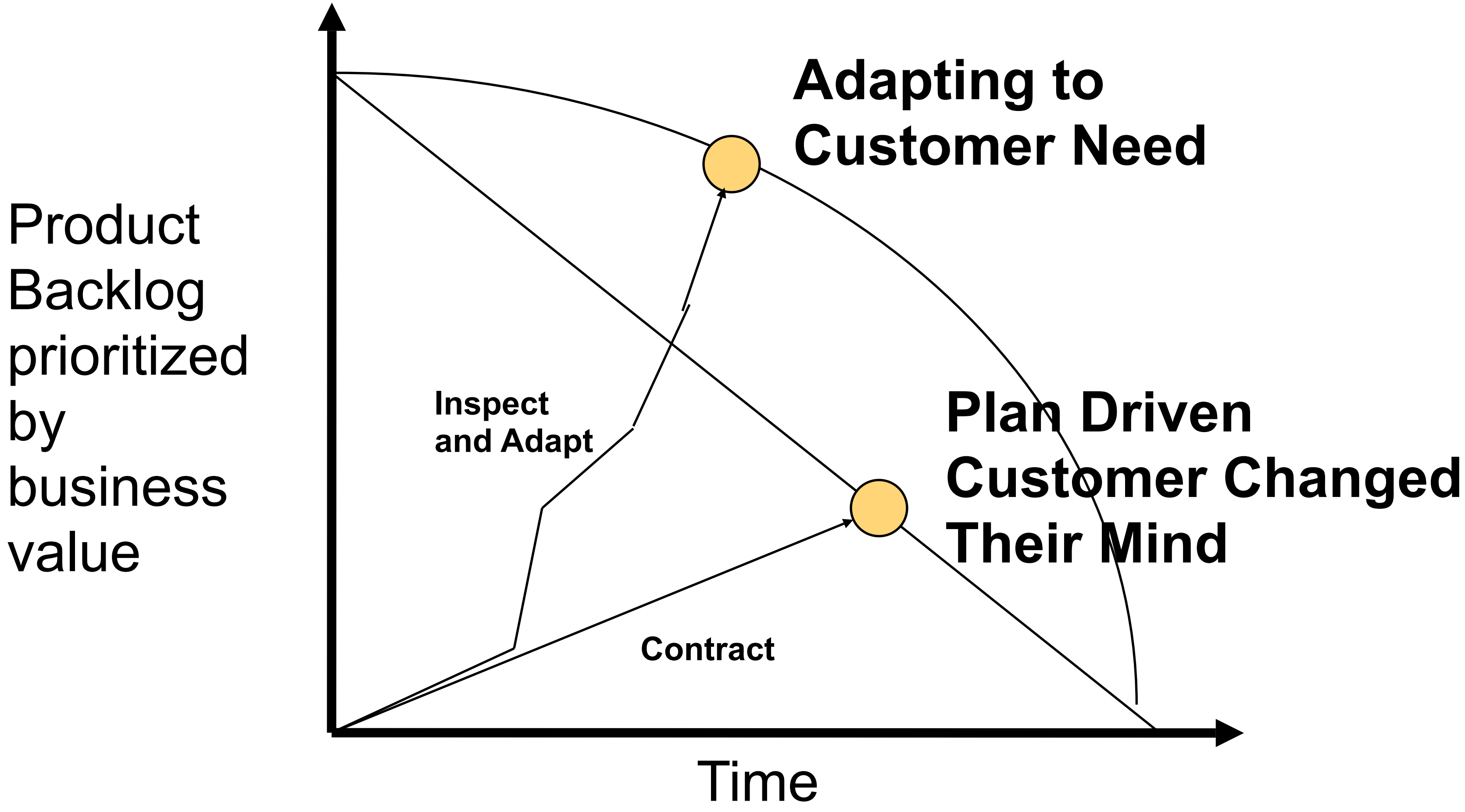
Avoidance of **Accountability**

Lack of **Commitment**

Fear of **Conflict**

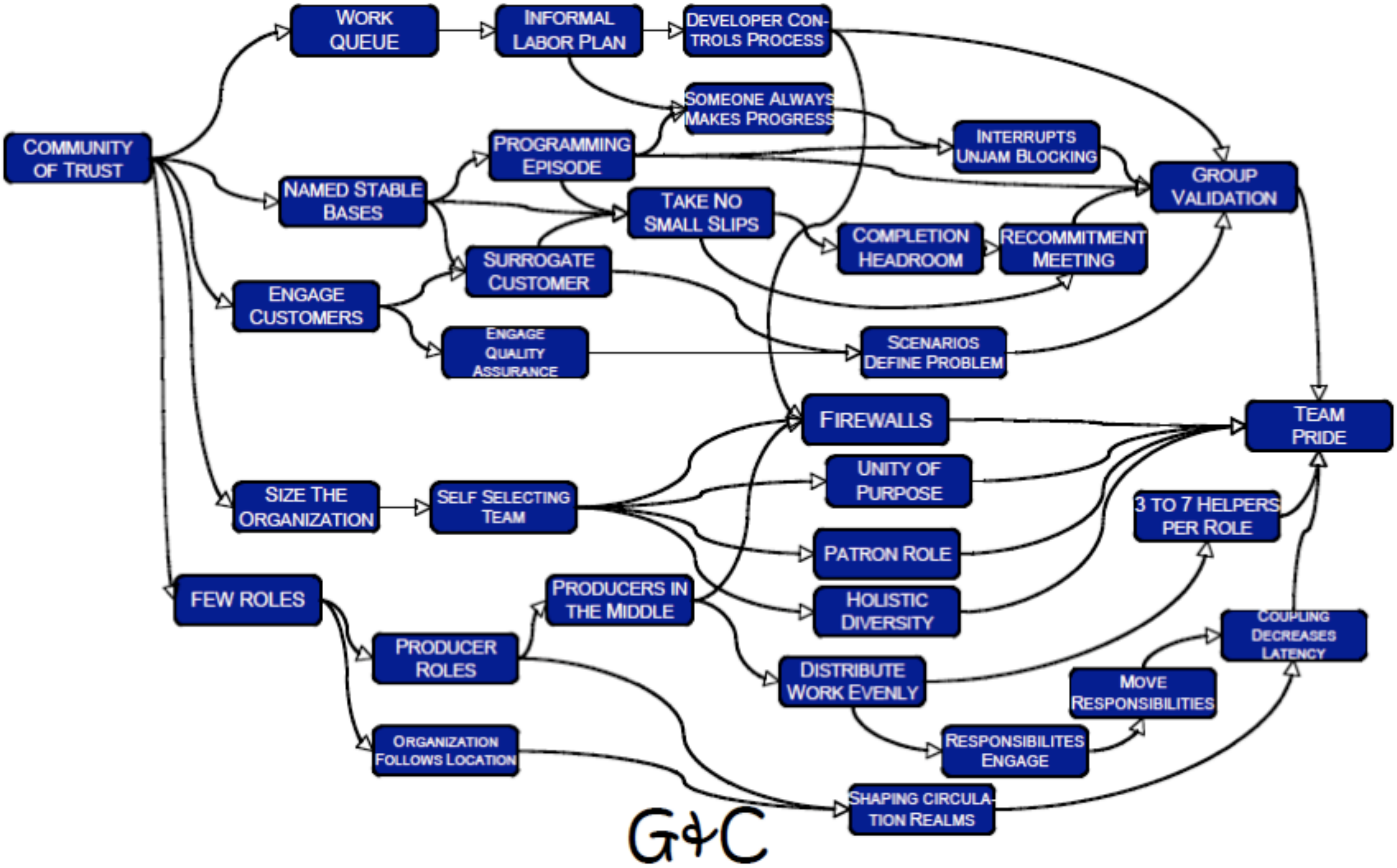
Absence of **Trust**

Business Value


Adapting to Customer Need
Product Backlog prioritized by business value
Inspect and Adapt
Plan Driven Customer Changed Their Mind
Contract
Time

© Jeff Sutherland 1993-2008

Scrum as patterns



Source: GertrudeandCope.com

© Jeff Sutherland 1993-2008

Self-Organization: Complex Adaptive Systems

- Self organization
  - Central planning will destroy it
- No single point of control
  - Command and control will crush it
- Interdisciplinary teams
  - Isolated activities and lack of transparency will cripple it
- Emergent behavior
  - Failure to remove impediments will ensure emergence of mediocrity
- Outcomes emerge in context
  - Empirical process requires inspect and adapt
- Team performance far greater than sum of individuals

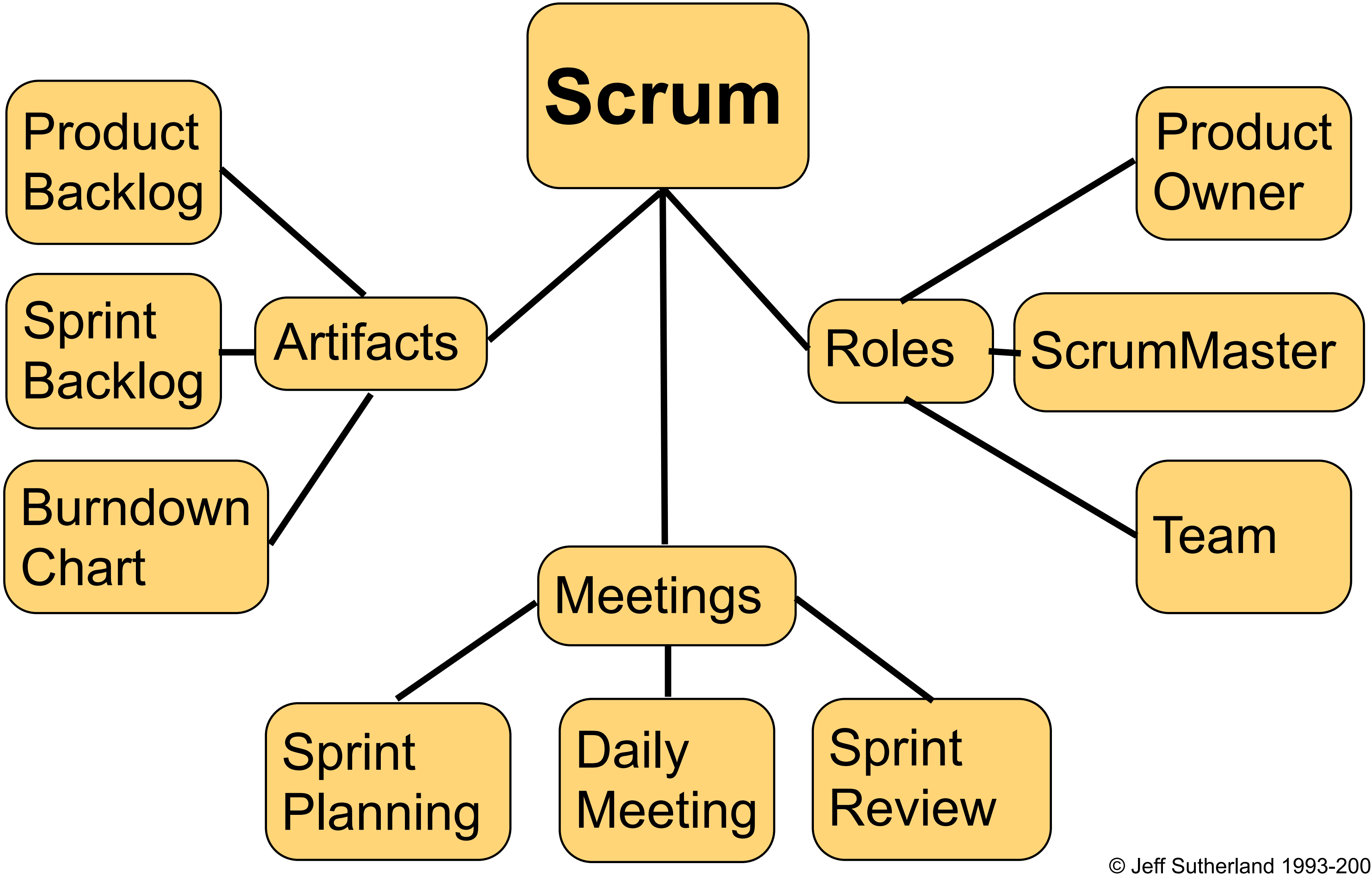
J. Sutherland, A. Viktorov, and J. Blount, ***Adaptive Engineering of Large Software Projects with Distributed/Outsourced Teams,*** in International Conference on Complex Systems, Boston, MA, USA, 2006.

Kaizen Mind

The ‘Toyota Way’ Is Translated for a New Generation of Foreign Managers
By Maring Fackler, New York Times, February 15, 2007

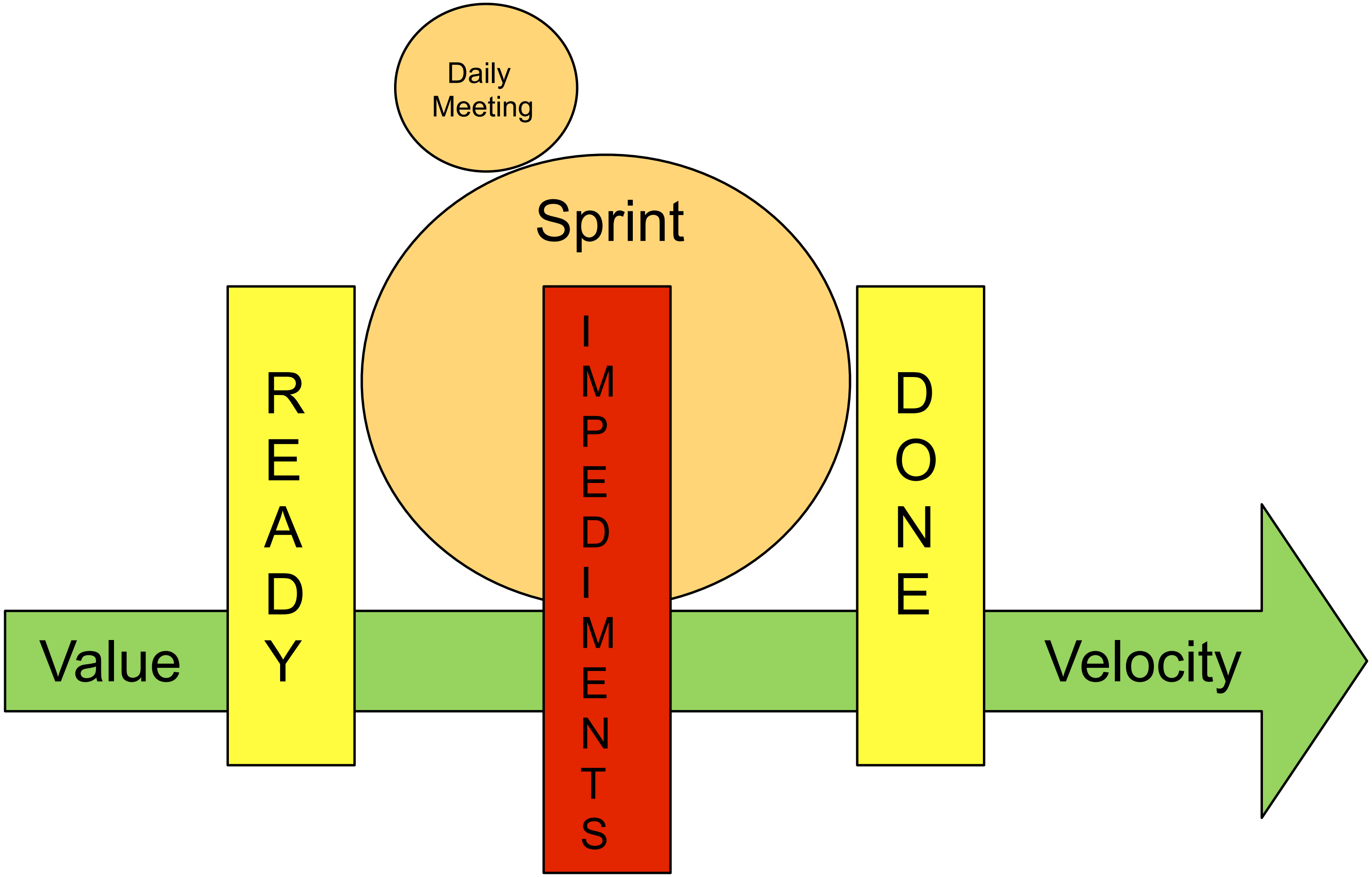
- 70% of improvement processes that require change fail, mainly due to lack of a sense of urgency among leadership - *John Kotter, Harvard Business School*
- “There is a sense of danger,” said Koki Konishi, who heads the Toyota Technical Skills Academy in Toyota City.
- “We must prevent the Toyota Way from getting more and more diluted as Toyota grows overseas.”
- We need “kaizen mind,” an unending sense of crisis behind the company’s constant drive to improve.

Scrum is a Simple Framework



© Jeff Sutherland 1993-2008

Scrum Dynamic Model



© Jeff Sutherland 1993-2008

Avoiding ScrumButt - Nokia Test Origins

Nokia Siemens Networks

- In 2005, Bas Vodde started training and coaching teams at Nokia Networks in Finland. The first Nokia test focused on Agile practices
  - jeffsutherland.com/scrum/basvodde2006_nokia_agile.pdf
- By 2007, Siemens had acquired Nokia Networks to form Nokia Siemens Networks with over 60,000 employees and 15 billion Euro in revenue. Bas Vodde moved to China to train Nokia Siemens Networks staff on Scrum and updated the Nokia Test to include Scrum practices.
- In 2007, Jeff Sutherland tuned the Nokia Test for Scrum Certification and in 2008 developed a scoring system.
  - agileconsortium.blogspot.com/2007/12/nokia-test.html

Question 1 - Iterations

- No iterations - 0
- Interations > 6 weeks - 1
- Variable length < 6 weeks - 2
- Fixed iteration length 6 weeks - 3
- Fixed iteration length 5 weeks - 4
- Fixed iteration 4 weeks or less - 10

Question 2 - Testing

- No dedicated QA - 0
- Unit tested - 1
- Feature tested - 5
- Features tested as soon as completed - 7
- Software passes acceptance testing - 8
- Software is deployed - 10

Question 3 - Agile Specification

- No requirements - 0
- Big requirements documents - 1
- Poor user stories - 4
- Good requirements - 5
- Good user stories - 7
- Just enough, just in time specifications - 8
- Good user stories tied to specifications as needed - 10

Question 4 - Product Owner

- No Product Owner - 0
- Product Owner who doesn’t understand Scrum - 1
- Product Owner who disrupts team - 2
- Product Owner not involved with team - 2
- Product owner with clear product backlog estimated by team before Sprint Planning meeting (READY) - 5
- Product owner with release roadmap with dates based on team velocity - 8
- Product owner who motivates team - 10

Question 5 - Product Backlog

- No Product Backlog - 0
- Multiple Product Backlogs - 1
- Single Product Backlog - 3
- Product Backlog clearly specified and prioritized by ROI before Sprint Planning (READY) - 5
- Product Owner has release plan based on Product Backlog - 7
- Product Owner can measure ROI based on real revenue, cost per story point, or other metrics - 10

Question 6 - Estimates

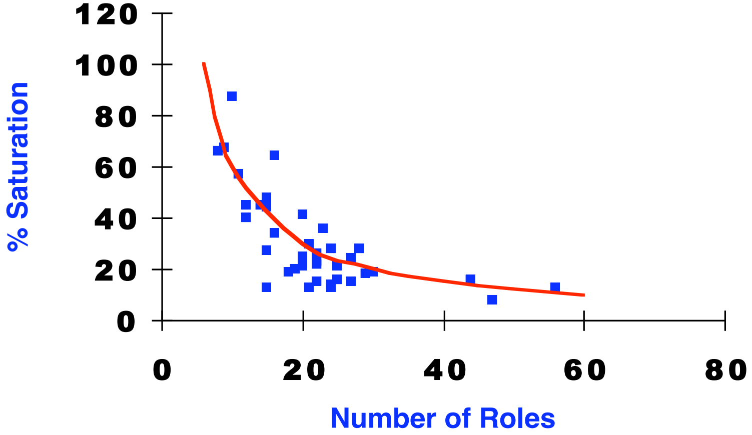
- Product Backlog not estimated - 0
- Estimates not produced by team - 1
- Estimates not produced by planning poker - 5
- Estimates produced by planning poker by team - 8
- Estimate error < 10% - 10

Question 7 - Burndown Chart

- No burndown chart - 0
- Burndown chart not updated by team - 1
- Burndown chart in hours/days not accounting for work in progress (partial tasks burn down) - 2
- Burndown chart only burns down when task in done - 4
- Burndown only burns down when story is done - 5
- Add 3 points if team knows velocity
- Add two point if Product Owner release plan based on known velocity

Question 8 - Team Disruption

- Manager or Project Leader disrupts team - 0
- Product Owner disrupts team - 1
- Managers, Project Leaders or Team leaders assigning tasks - 3
- Have Project Leader and Scrum roles - 5
- No one disrupting team, only Scrum roles - 10



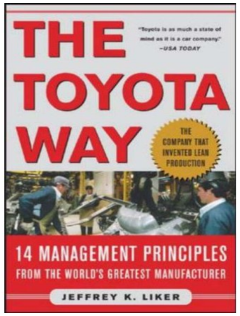
Organizational Patterns of Agile Software Development by Coplien and Harrison (2004)

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.

Can failure help us?

How many different types of light bulbs did Edison build before he got one to work?


THE TOYOTA WAY
THE COMPANY THAT INVENTED LEAN PRODUCTION
14 MANAGEMENT PRINCIPLES FROM THE WORLD'S GREATEST MANUFACTURER
JEFFREY K. LIKER

Toyota Way: Learn by Doing

Fujio Cho, Board Chairman

- We place the highest value on actual implementation and taking action. *Agile Principle #1*
- There are many things one doesn’t understand, and therefore we ask them why don’t you just go ahead and take action; try to do something? *Agile Principle #3, #11*
- You realize how little you know and you face your own failures and redo it again and at the second trial you realize another mistake … so you can redo it once again. *Agile Principle #11, #12*
- So by constant improvement … one can rise to the higher level of practice and knowledge. *Agile Principle #3*

"Anyone who has never made a mistake has never tried anything new." Albert Einstein

How we invented Scrum:

Alan Kay's innovation strategy at Xerox Parc

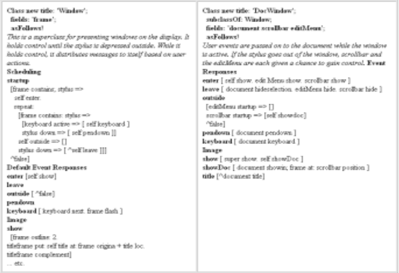
Personal Workstation

Mouse (SRI)

Ethernet

Windows Interface

Laser Printer



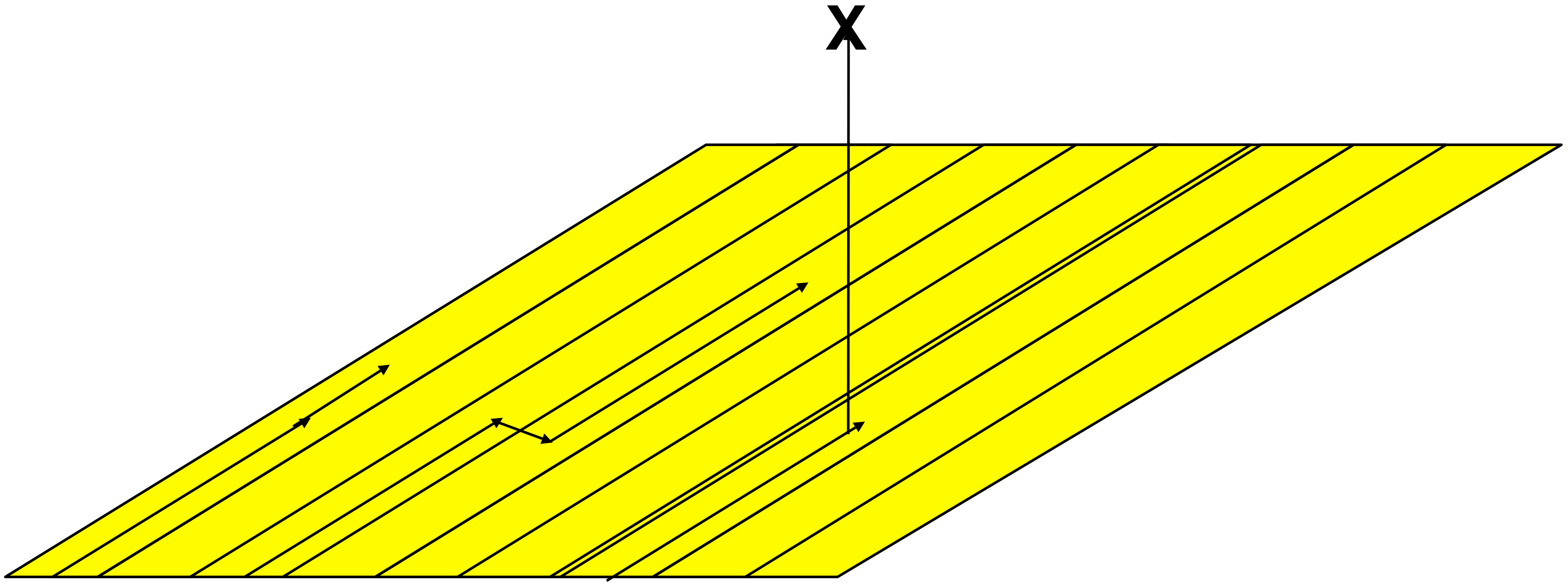
```
Class new title: 'Window';
 fields: 'frame';
 asFollows!
This is a superclass for presenting windows on the display. It
holds control until the stylus is depressed outside. While it
holds control, it distributes messages to itself based on user
actions.
Scheduling
startup
 [frame contains: stylus =>
   self enter.
   repeat:
    [frame contains: stylus =>
      [keyboard active => [ self keyboard ]
      stylus down => [ self pendown ]]
    self outside => []
    stylus down => [ ^self leave ]]]
 ^false]
Default Event Responses
enter [self show]
leave
outside [ ^false]
pendown
keyboard [ keyboard next. frame flash ]
Image
show
 [frame outline: 2.
titleframe put: self title at: frame origina + title loc.
titleframe complement]
... etc.
```

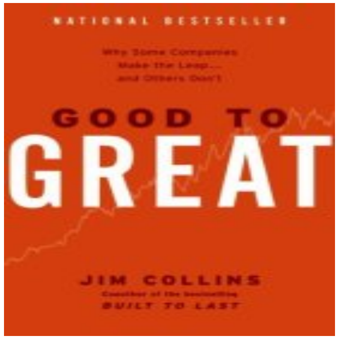
```
Class new title: 'DocWindow';
 subclassOf: Window;
 fields: 'document scrollbar editMenu';
 asFollows!
User events are passed on to the document while the window
is active. If the stylus goes out of the window, scrollbar and
the editMenu are each given a chance to gain control. Event
Responses
enter [ self show. edit Menu show. scrollbar show ]
leave [ document hideselection. editMenu hide. scrollbar hide ]
outside
 [editMenu startup => []
 scrollbar startup => [self showdoc]
 ^false]
pendown [ document pendown ]
keyboard [ document keyboard ]
Image
show [ super show. self showDoc ]
showDoc [ document showin; frame at: scrollbar position ]
title [^document title]
```

Smalltalk

Alan Kay’s Innovation Strategy

- Incremental - NO
- Cross Discipline - NYET
- Extreme data points - ONLY LOOK AT THIS!





NATIONAL BESTSELLER

Why Some Companies Make the Leap... and Others Don't

GOOD TO GREAT

JIM COLLINS

Coauthor of the bestselling BUILT TO LAST

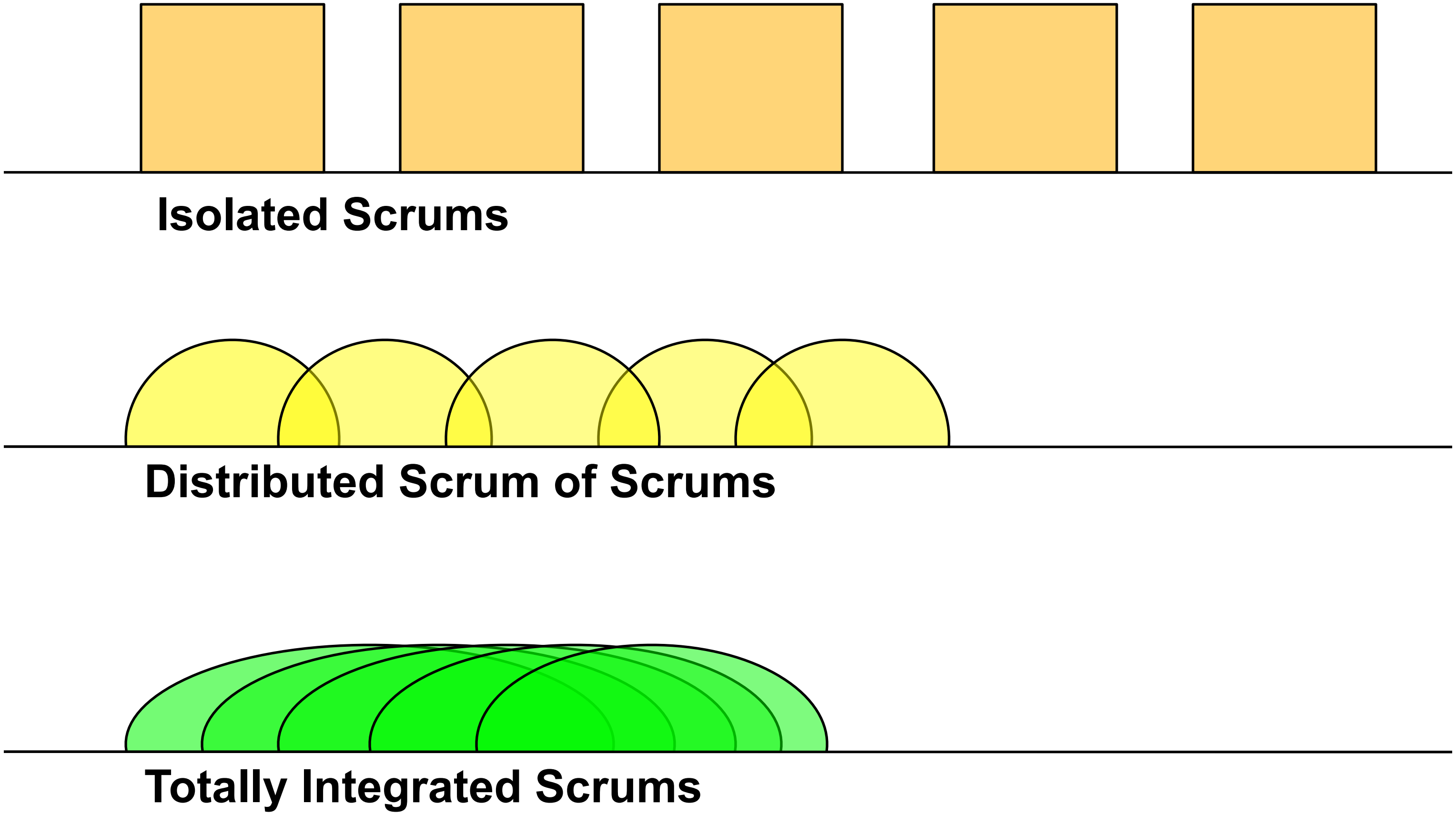
Out of the Box

- Scrum looked at projects that were off the chart
  - IBM surgical team
  - Takeuchi and Nonaka
  - Borland Quattro Project
- *Scrum: A Pattern Language for Hyperproductive Software Development*
  - By M. Beedle, M. Devos, Y. Sharon, K. Schwaber, and J. Sutherland. In Pattern Languages of Program Design. vol. 4, N. Harrison, Ed. Boston: Addison-Wesley, 1999, pp. 637-651.
- Going from good to great means Toyota or better.

Case Study: Scrum and XP

- The first Scrum used all the XP engineering practices, set-based concurrent engineering, and viewed software development as maintenance, not manufacturing.
- Most high performance teams use Scrum and XP together.
- It is hard to get a Scrum with extreme velocity without XP engineering practices.
- You cannot scale XP without Scrum.
- Example 1: Anatomy of a failed project - SirsiDynix
  - ScrumButt
- Example 2: Xebia ProRail project
  - Pretty Good Scrum

Distributed/Outsourcing Styles



Isolated Scrums

Distributed Scrum of Scrums

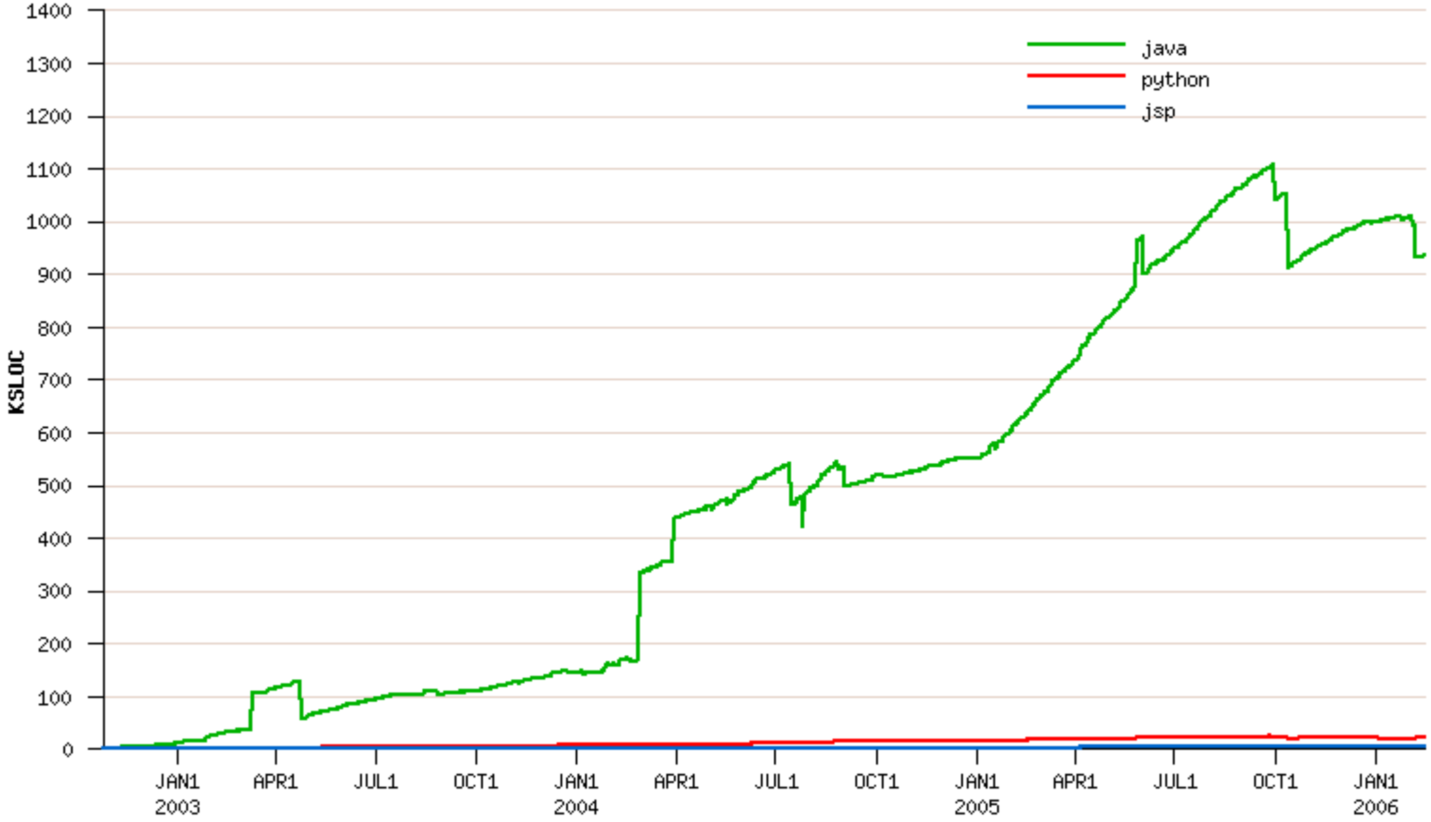
Totally Integrated Scrums

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.

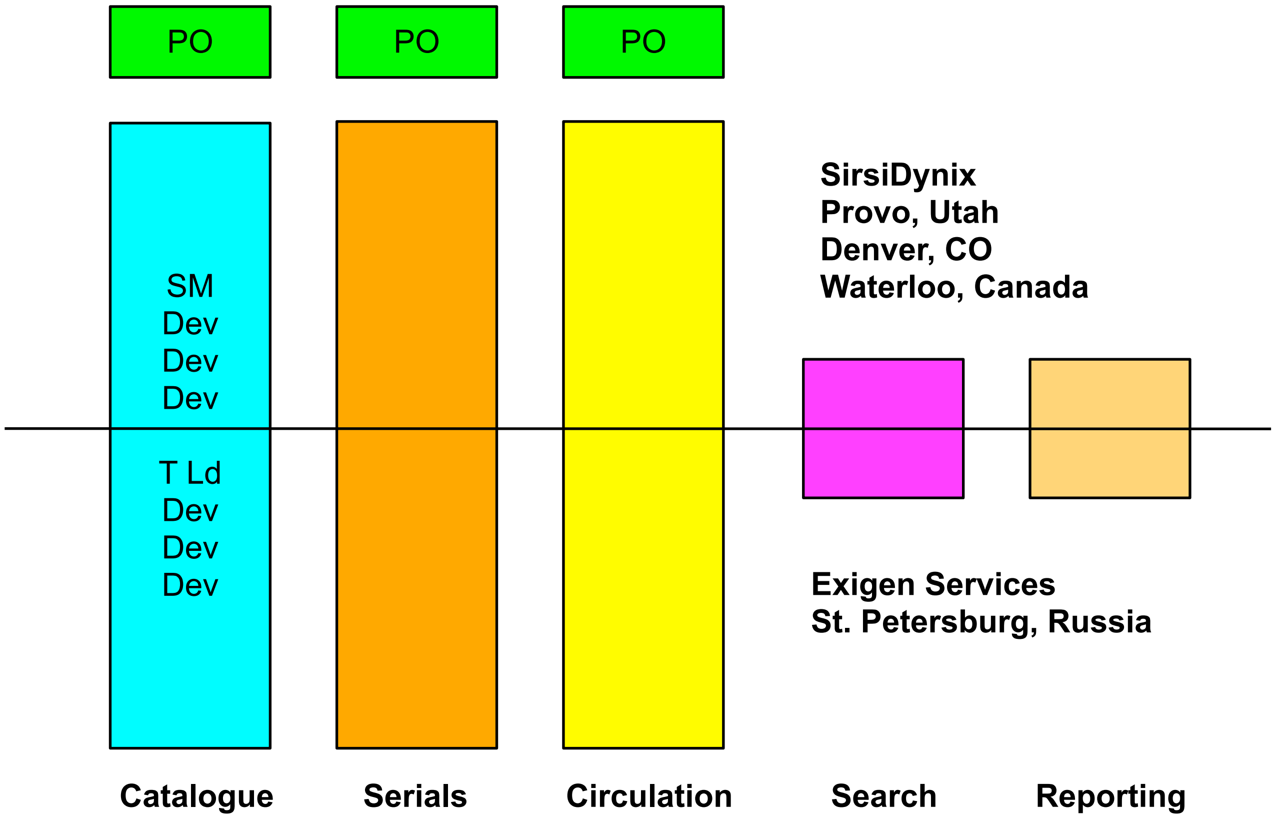
SirsiDynix - Anatomy of a failed project

- Over a million lines of Java code



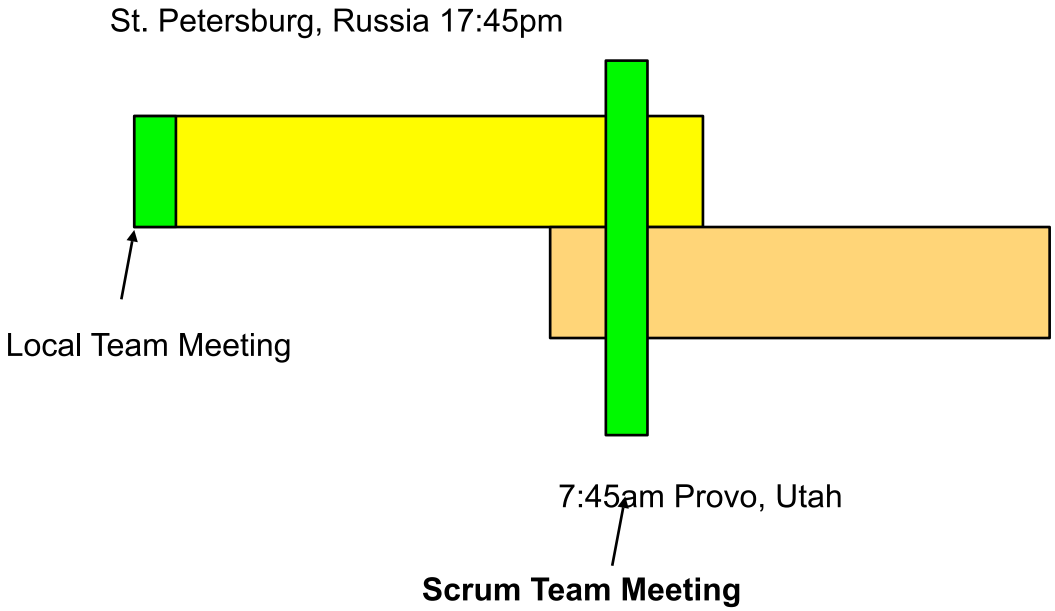
SirsiDynix Distributed Scrum

- 56 developers distributed across sites

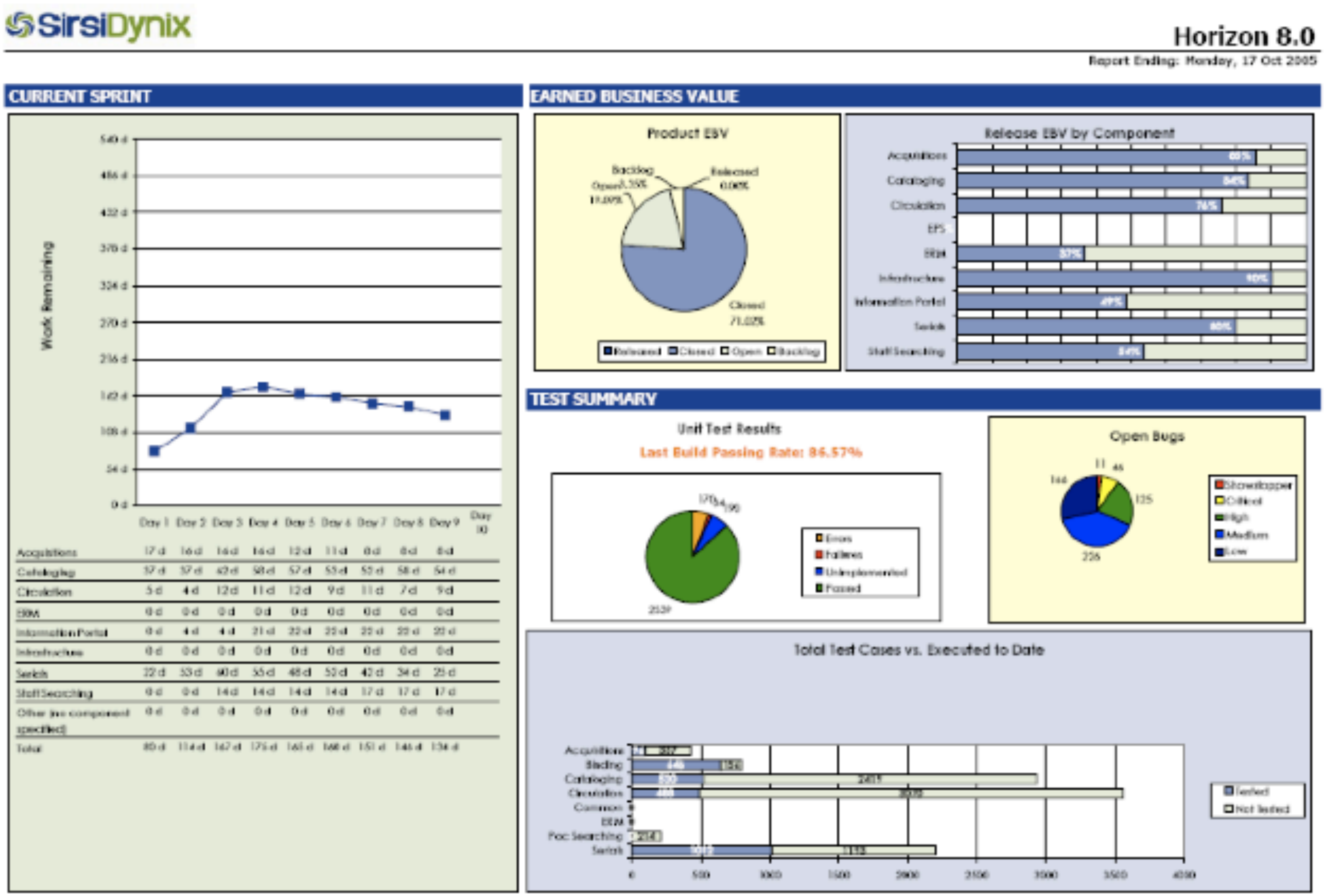


SirsiDynix Distributed Scrum

- Scrum daily meetings



SirsiDynix Distributed Scrum



SirsiDynix

Horizon 8.0

Report Ending: Monday, 17 Oct 2005

CURRENT SPRINT

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Acquisitions	17 d	16 d	16 d	16 d	12 d	11 d	8 d	8 d	8 d	
Cataloging	57 d	57 d	62 d	58 d	57 d	53 d	53 d	58 d	54 d	
Circulation	5 d	4 d	12 d	11 d	12 d	9 d	11 d	7 d	9 d	
ERM	0 d	0 d	0 d	0 d	0 d	0 d	0 d	0 d	0 d	
Information Portal	0 d	4 d	4 d	21 d	22 d	22 d	22 d	22 d	22 d	
Infrastructure	0 d	0 d	0 d	0 d	0 d	0 d	0 d	0 d	0 d	
Serials	22 d	53 d	60 d	55 d	48 d	53 d	42 d	34 d	25 d	
Staff Searching	0 d	0 d	14 d	14 d	14 d	14 d	17 d	17 d	17 d	
Other (no component specified)	0 d	0 d	0 d	0 d	0 d	0 d	0 d	0 d	0 d	
Total	80 d	114 d	167 d	175 d	165 d	168 d	151 d	146 d	136 d	

EARNED BUSINESS VALUE

TEST SUMMARY

Unit Test Results

Last Build Passing Rate: 86.57%

© Jeff Sutherland 1993-2008

Velocity in Function Points/Dev month

	Scrum[1]	Waterfall[1]	SirsiDynix[2]
Person Months	54	540	827
Lines of Java	51,000	58,000	671,688
Function Points	959	900	12673
Function Points per Dev/Mon	17.8	2.0	15.3

1. M. Cohn, User Stories Applied for Agile Development. Addison-Wesley, 2004
2. J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii,

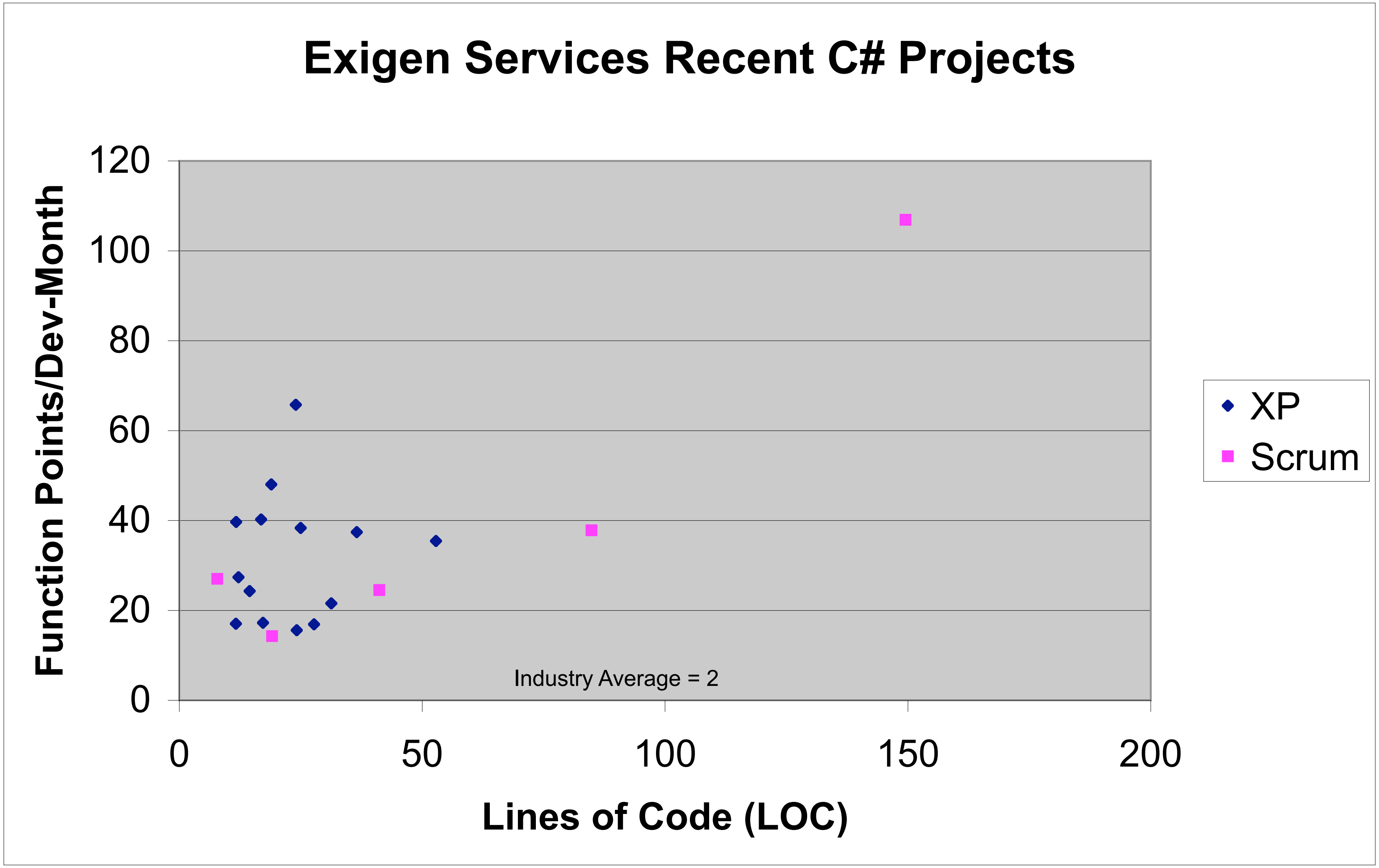
SirsiDynix Challenges

- 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, did not have equal talent across teams
- No face to face meetings
- Low test coverage
- Poor refactoring practice
- 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?

Russian projects velocity data suggests high velocity is not an accident



© Jeff Sutherland 1993-2008

Going Fast is Important

- If you can’t go 10 times faster than the competition it may not matter how good your quality is.
- It is easier to fix quality than it is to get hyperproductive.
- Applying lean principles to a high velocity implementation will make quality go up while simultaneously making velocity even faster.
- Applying quality processes to a non-lean implementation will usually make it go slower.

SirsiDynix Opportunity

- A prospective study should be done
  - full XP technical practices
  - multiple projects
  - meet or exceed SirsiDynix velocity
  - ensure quality levels in the top 1% of the software industry.

Setting up 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

Can you have a distributed team?

- Distributed software development and offshoring eliminates collocation which has been shown to double developer productivity compared to non-colocated teams {Teasley, 2000}.
- Much of the lost productivity of distributed teams is caused by their inability to function as one team.
- Research on characteristics of emergency response teams {Plotnick, 2008} shows that in almost all cases distributed subgroups function as independent entities with conflict between groups.
- However, in rare cases a larger team identity is formed that subsumes the subgroups.

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

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.*

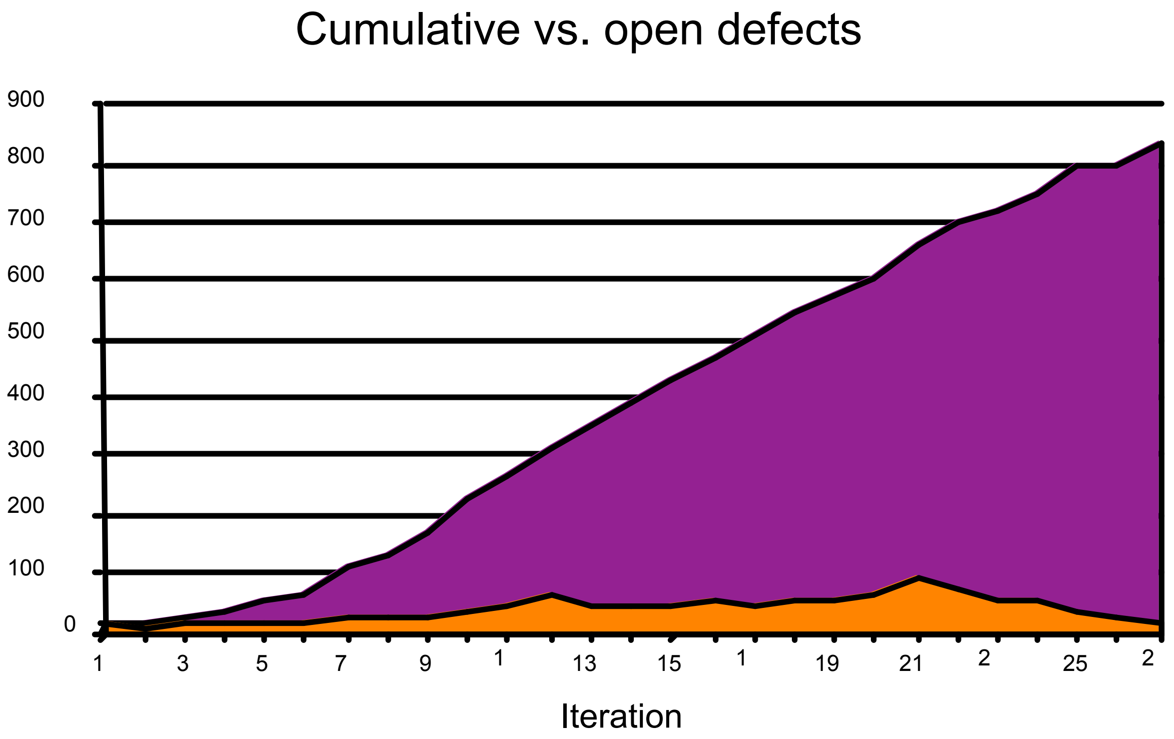
ProRail PUB Example

- ProRail rescued a failed waterfall project to build a new scheduling system and automated railway station signs at all Netherlands railway stations
- An 8 person Dutch Scrum team started the project and established local velocity.
- Xebia’s India subsidiary sent 8 people to the Netherlands and two teams were formed. Each team was 4 Dutch and 4 Indian programmers.
- After establishing local velocity at 5 times other waterfall vendors on the project, the Indian half of each team went back to India.

ProRail Definition of Done

- Scrum teams run all XP practices inside the Scrum including intensive pair programming.
- The customer completes acceptance testing on all features during each Sprint.
- Done at the end of the Sprint means customer has accepted the code as ready for production.
- Defect rates are less than 1 per 1000 lines of code and steadily getting lower.

ProRail Defect Tracking



- Defect rate gets lower and lower as code base increases in size
- 95% of defects found inside iteration are eliminated before the end of the iteration

Team Characteristics

- TDD, pair programming, continuous integration. Same tools and techniques onshore and offshore.
- Daily Scrum meeting of team across geographies.
- SmartBoards, wikis, and other tools used to enhance communication.
- Indians say it feels exactly the same in India as it does in Amsterdam. They do the same thing in the same way.
- Xebia CTO has decided to use this model on all projects because it provides (counterintuitively) better customer focus and all other metrics are the same onshore or offshore.

Resolving Cultural Differences

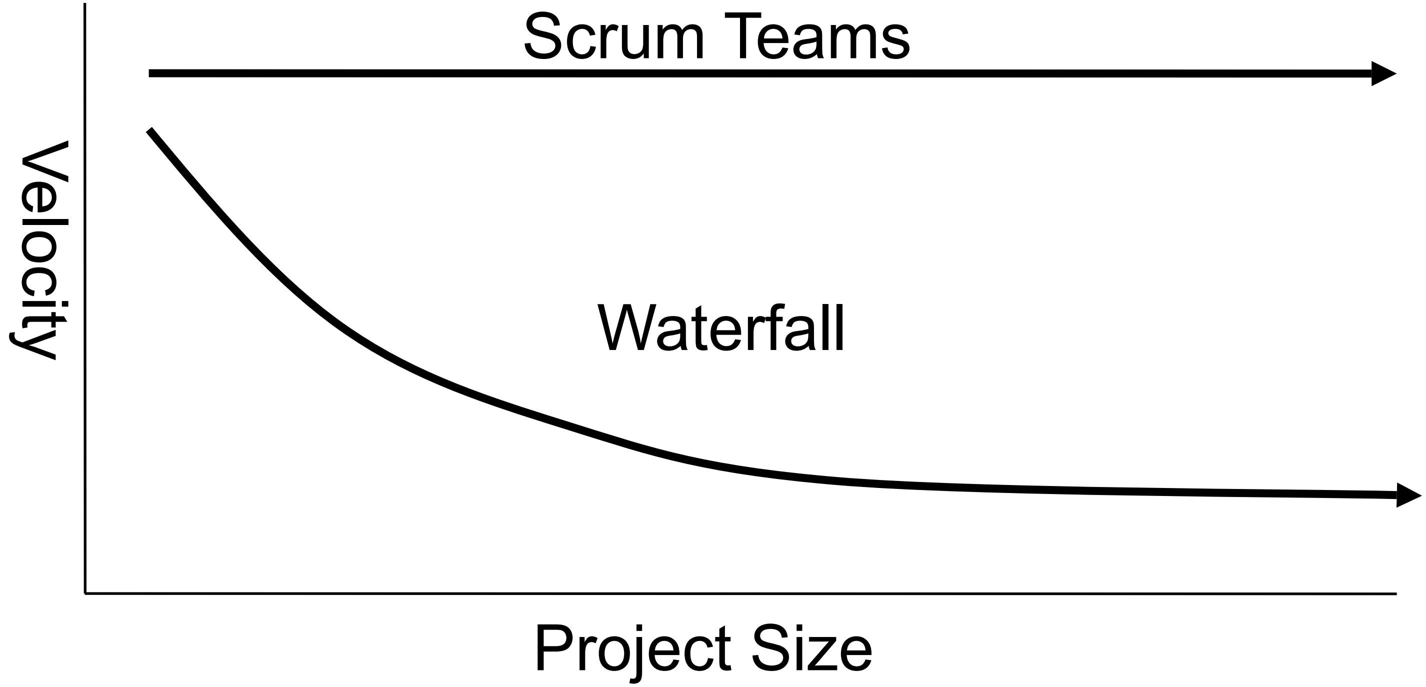
- One of the teams had local velocity decrease after distributing the team.
- Root cause analysis indicated the Indians were waiting for the senior Indian developer to tell them what to do.
- The same day this was determined, the Dutch ScrumMaster became a team member and the lead Indian developer became the ScrumMaster with the goal of eliminating the impediment.
- Distributed velocity immediately went up to previously established local velocity.

Dutch Velocity vs. Russian Velocity

	SirsiDynix[2]	**Xebia[3]**
Person Months	827	125
Lines of Java	671,688	**100,000**
Function Points	12673	**1887**
Function Points per Dev/ Mon	15.3	**15.1**

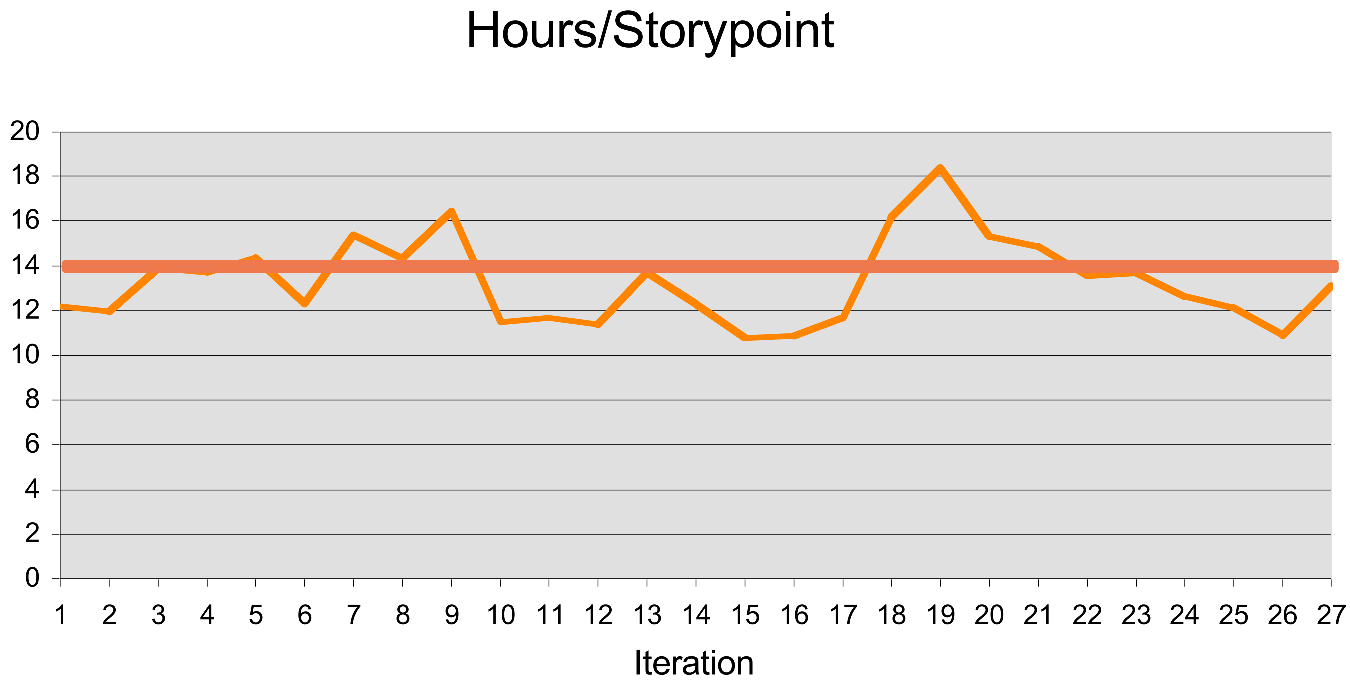
1. M. Cohn, User Stories Applied for Agile Development. Addison-Wesley, 2004
2. J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii,
3. J. Sutherland, G. Schoonheim, E. Rustenburg, M. Rijk. Fully Distributed Scrum: The Secret Sauce for Hyperproductive Outsourced Development Teams. Agile 2008, Toronto, Aug 4-8 (submission, preliminary data)

Linear Scalability of Large Scrum Projects



- J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii, 2007.
- J. Sutherland, C. Jacobson, and K. Johnson, "Scrum and CMMI Level 5: A Magic Potion for Code Warriors!," in Agile 2007, Washington, D.C., 2007.

Linear scalability



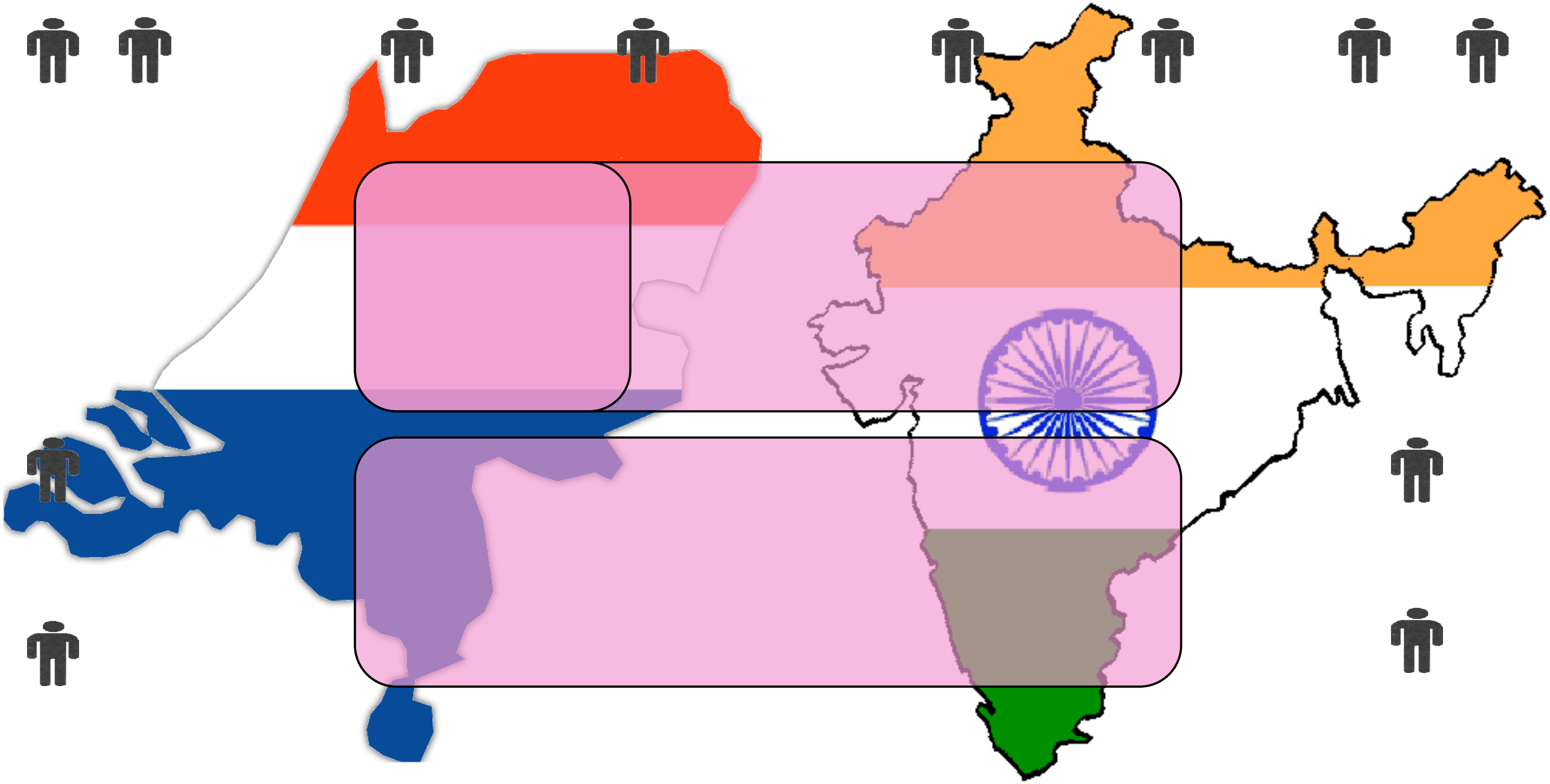
Conclusion

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

Fully Distributed Scrum has more value then localized Scrum.

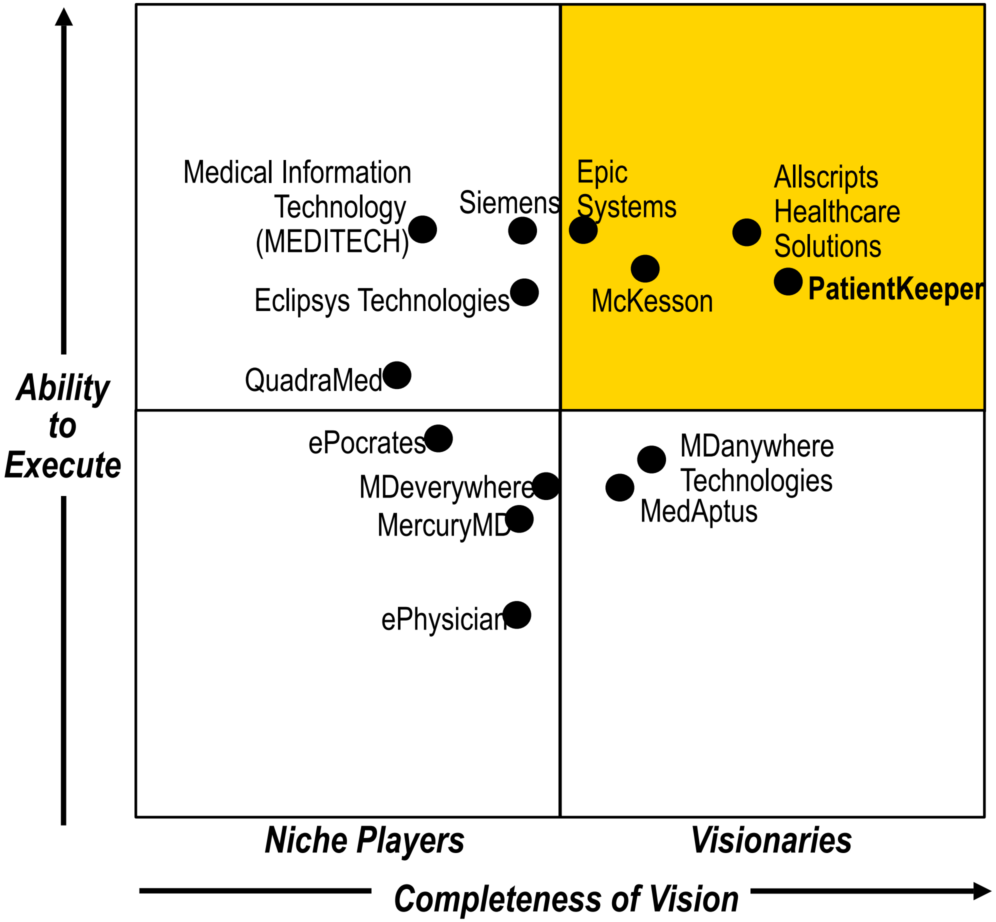
All Xebia projects of more than a few people are fully distributed today.

Fully Distributed Scrum



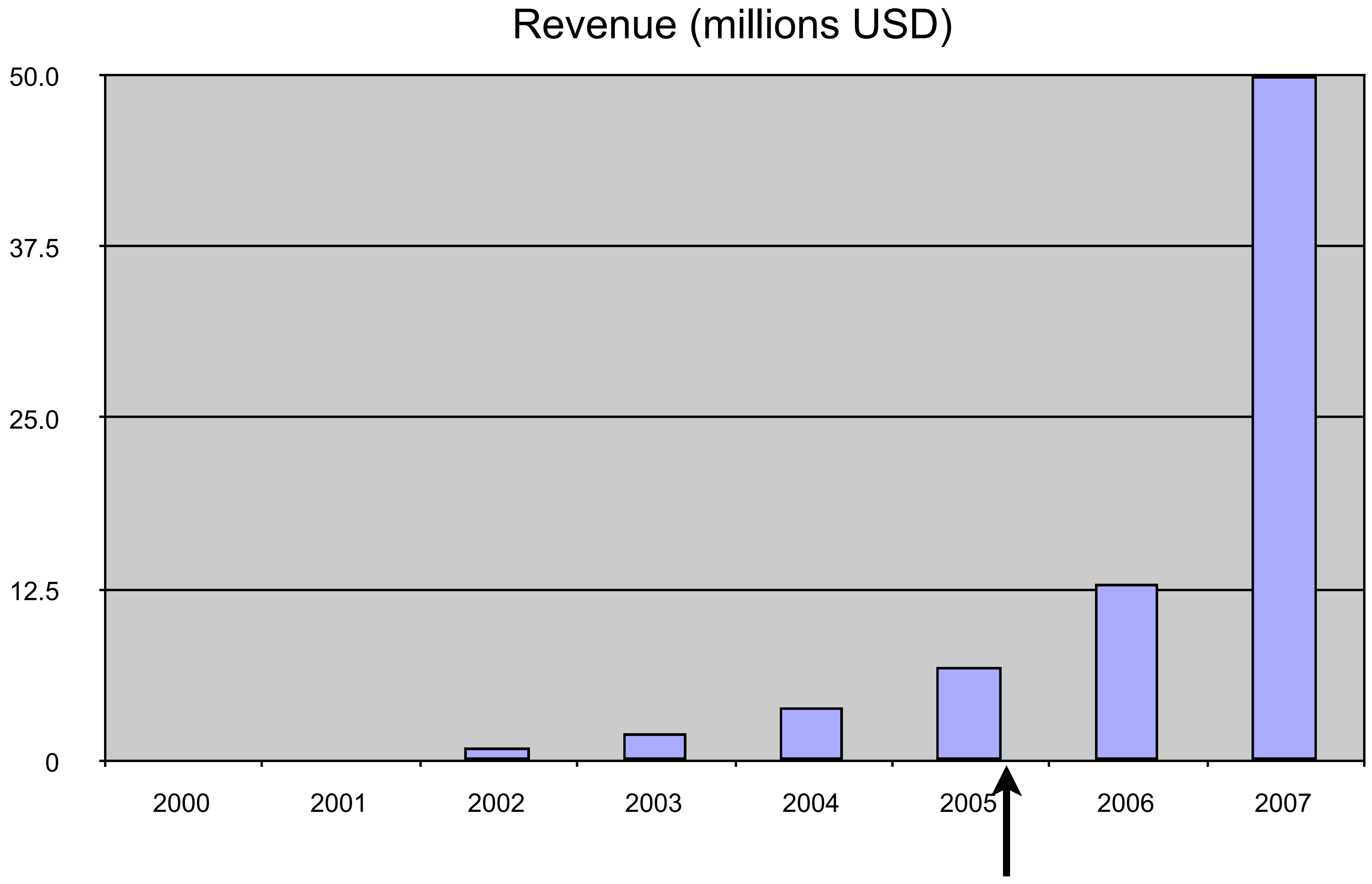
© Jeff Sutherland 1993-2008

Avoid Thrashing ...



I find that the vast majority of organizations are still trying to do too much stuff, and thus find themselves thrashing. The only organization I know of which has really solved this is PatientKeeper. **Mary Poppendieck**

PatientKeeper Revenue

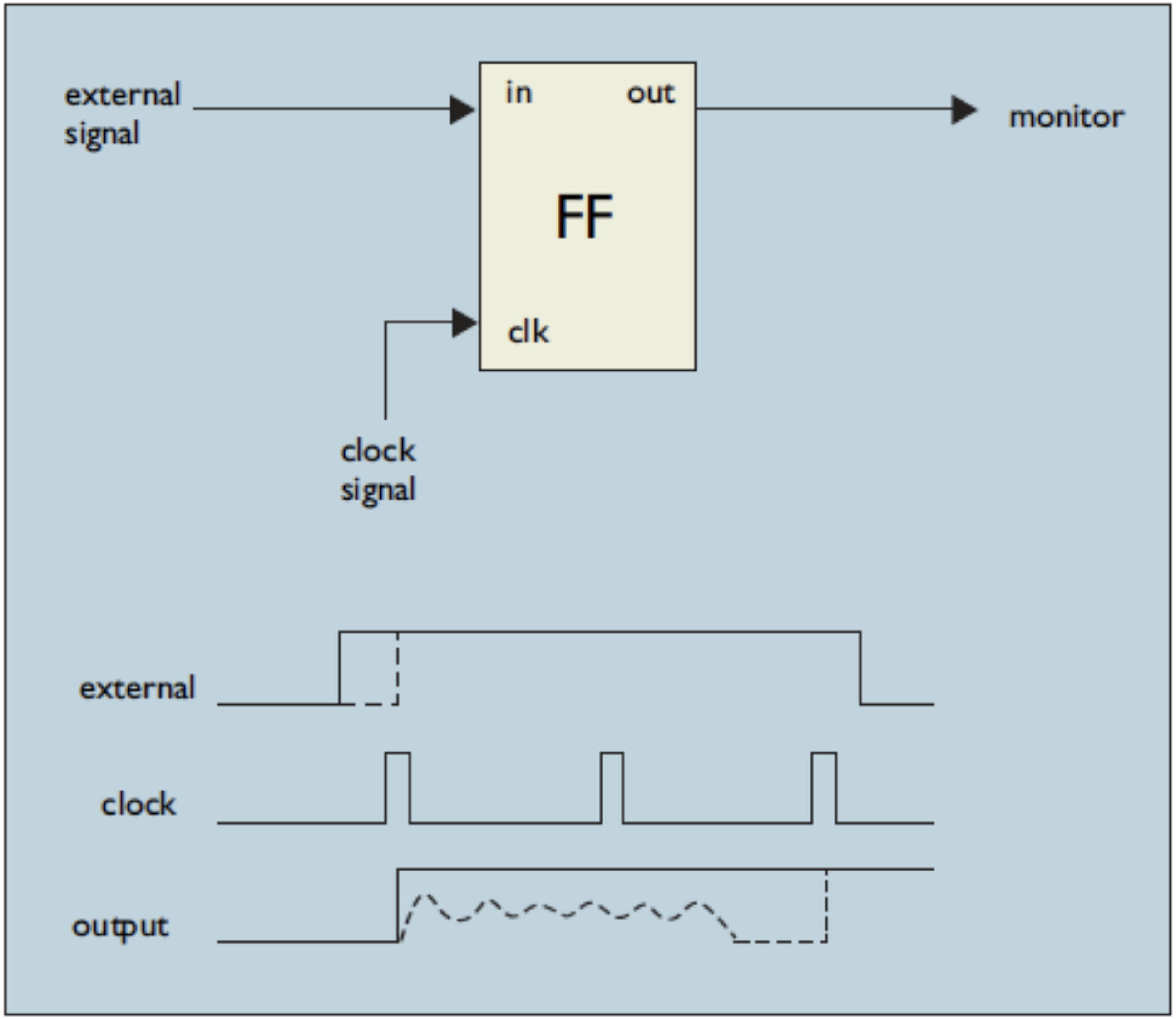


Thrashing ended

Cosmic Stopping Problem

- Early computers crashed randomly. It was attributed to “cosmic rays” that corrupted memory.
- Computer logic circuits occasionally locked up, requiring a complete restart of the computer. These lockups could cause the loss of considerable work on long computations. Worse, they made machines unreliable for real-time, safety-critical applications.
- These lockups never occurred when the interrupts were off. (No disruption!)

Denning, Peter. The Choice Uncertainty Principle. Computer 50:11, p. 9, November 2007



If Flip Flop is in indeterminate state at clock signal bad things happen.

Metastable State

- At less than 200MHz modern flipflops have zero chance of metastable states
- At 300MHz they disrupt computer every two weeks
- At 400MHz there is a problem every three minutes
- Solution was to force CPU not to read unless state was stable

Extremely valuable, in some situations, to know which state you are in

- Example: two people passing on a sidewalk not knowing which way to pass. They must stop until they decide

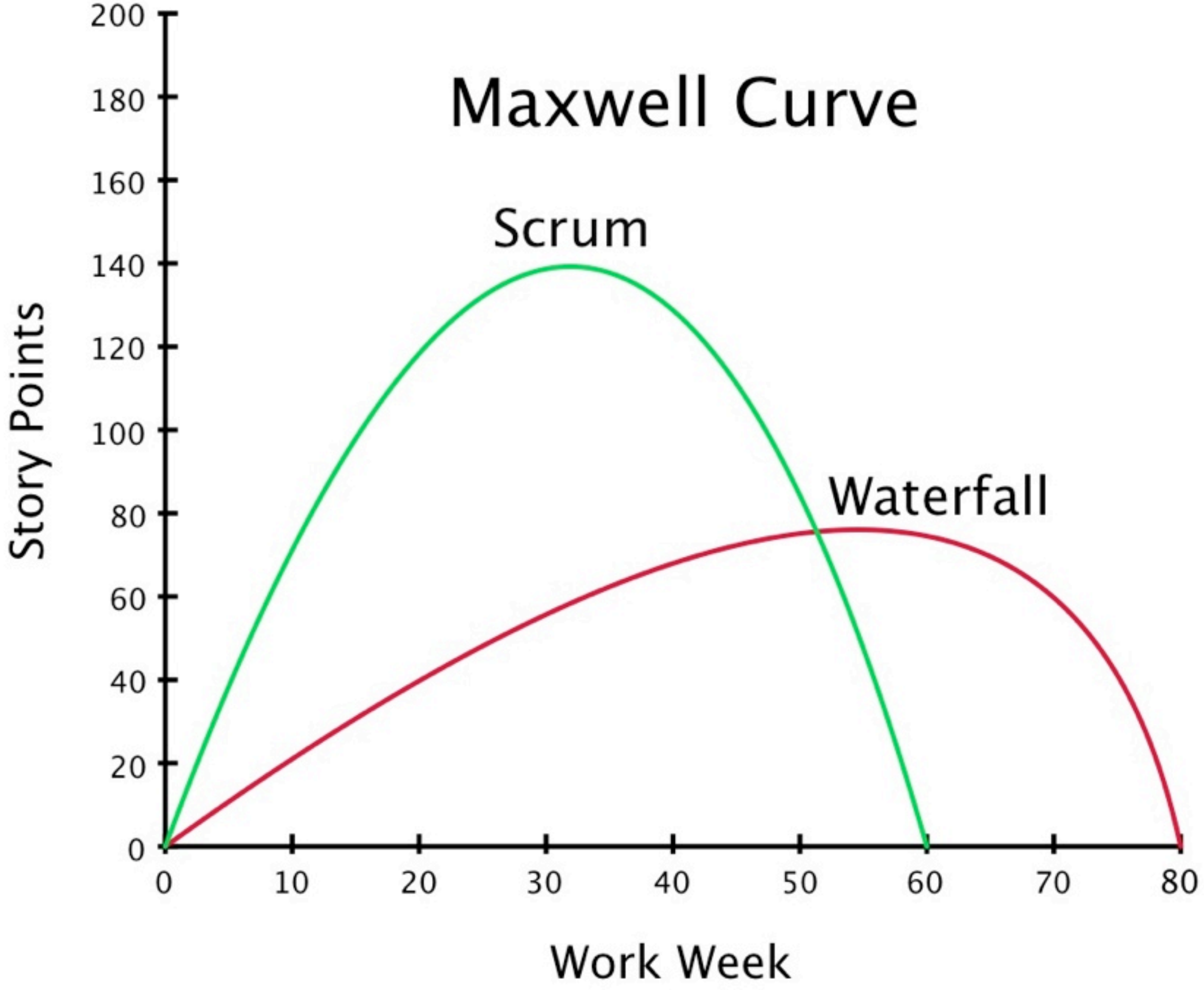
Software Metastable State

- If velocity is 2 Function Points per developer-month and release time is one year, there is almost zero chance of finding a metastable state that causes a crash mid-year
- If velocity is 16 Function Points per developer-month and there are 45 releases per year, there are 16*45 = 720 more times to run into something broken.
- Every part of Scrum, every piece of every meeting is involved in checking for metastable states. Stopping and stabilizing the state before moving on is generally best practice.

Choice Uncertainty Principle is a Cosmic Problem

- It occurs at every level of the universe (Heisenberg Uncertainty Principle)
- Well-run Scrums handle this at every step
  - Don’t accept backlog that is not ready
  - Minimize work-in-progress
  - Stop the line when bad things happen and fix it so it can never happen again
  - If it is not “Done” put it back on the product backlog
- Forceful stopping is challenging

Goal - double output, more than double quality, cut workload in half

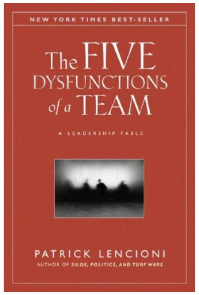


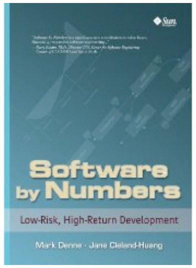
www.openviewventurepartners.com

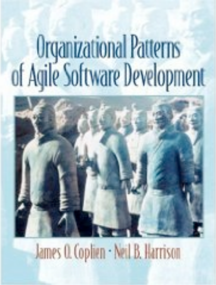
Questions?

Books on Scrum drivers …

- Human values
  - Patrick Lencioni. *Five Dysfunctions of a Team: A Leadership Fable.* Jossey-Bass, 2002
- Business value
  - Mark Denne and Jane Cleland-Huang. *Software by Numbers.* Prentice Hall, 2003.
- Patterns
  - Jim Coplien and Neil Harrison. *Organizational Patterns of Agile Software Development.* Prentice Hall 2004.
- Self-Organization
  - Taiichi Ohno and Norman Bodek. *Toyota Production System: Beyond Large Scale Production.* Productivity Press, 1988.
- Kaizen Mind
  - John Kotter. *A Sense of Urgency.* Harvard Business School Press, 2008.




Software by Numbers
Low-Risk, High-Return Development
Mark Denne · Jane Cleland-Huang


Organizational Patterns of Agile Software Development
James O. Coplien · Neil B. Harrison