Pretty Good Scrum

Fully Distributed Scrum: The Secret Sauce for Hyperproductive Outsourced Development 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, Juske Bank, BEC, Camp Scrum, DotWay AB, Ultimate Software, Scrum Training Institute, AtTask, Intronix, Loyalty Lab, Version One, OpenView Labs, Central Desktop, Open-E, Zmags

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 Institute

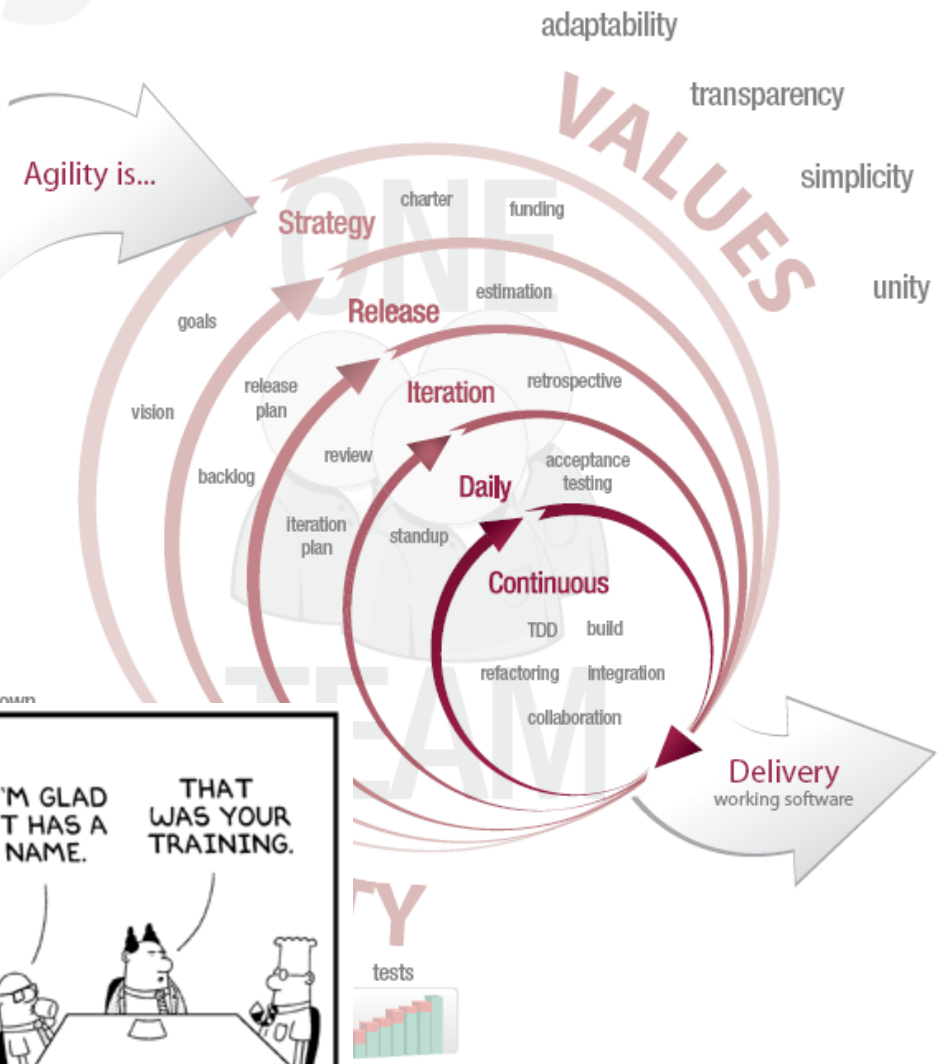
openview venture partners

SCRUM TRAINING INSTITUTE

Agile Development

ScrumButt

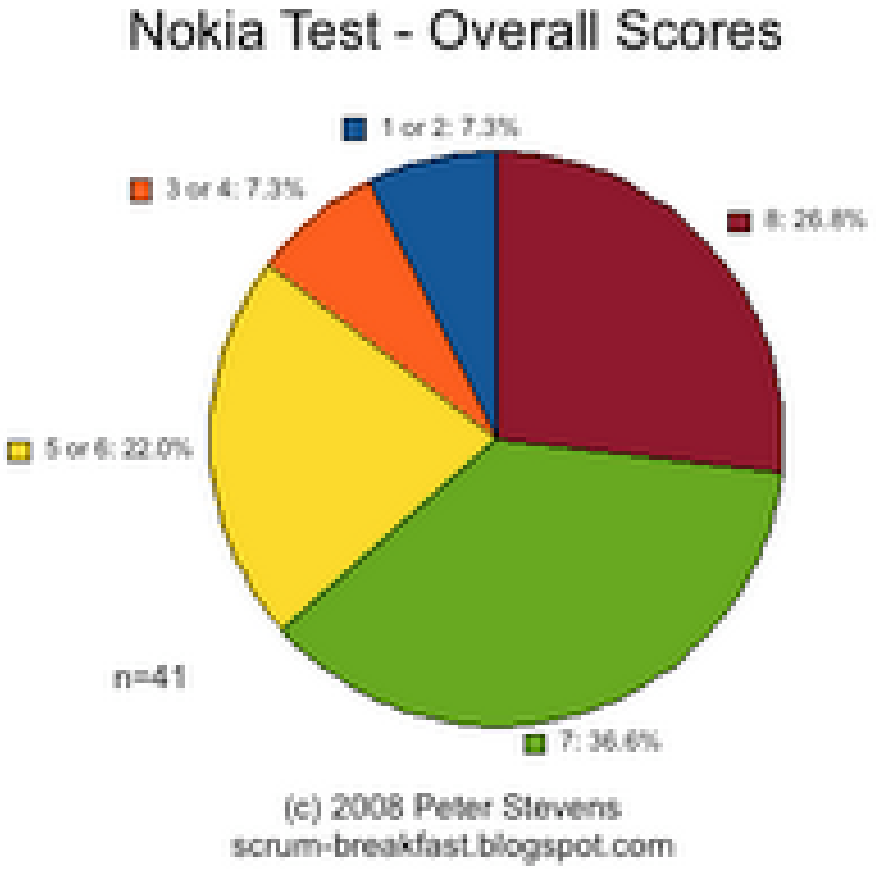
© Scott Adams, Inc./Dist. by UFS, Inc.



Accelerate Success

VERSIONONE
Simplifying Software Delivery

ScrumButt
74% of Scrum teams



Another way to measure ScrumButt

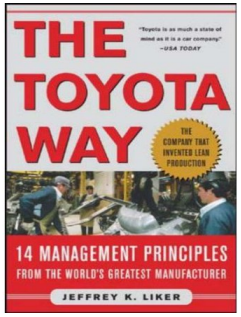
- Excellent Scrum - annual revenue up 400%
  - PatientKeeper
  - Others in Scandinavia I can't talk about
- Good Scrum - revenue up 300%
  - Companies in Scandinavia I can't talk about
- Pretty Good Scrum - revenue up 150% - 200%
  - Systematic Software Engineering - 200%
  - Google - 160%
- ScrumButt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

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?



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

BusinessWeek

Toyota's quality slips, again

David Welch, 17 October 2007

It's been a rough week for Toyota. First, the company gets slapped by Consumer Reports, which said that Toyota's quality slipped so much that it will no longer recommend every car just because it has a loopy "T" on the hood. CR won't recommend the V-6 powered Toyota Camry sedan nor the Tundra pickup because their quality results were below average. Overall, Toyota slipped in CR's latest survey. Its namesake brand now ranks fifth, behind Honda, Acura, Scion and Subaru. The next day, Toyota said it would recall 470,000 cars in Japan.

Follow the money: Learn from venture capital investments

- Invest only in Agile projects
  - 1 hyperproductive company out of 10 is good enough to meet investment goals
  - Invest in Scrum training could get 2 hyperproductive
- Invest only in market leading, industry standard processes – this means Scrum and XP
- Ensure teams implement basic Scrum practices
  - Everyone must pass Nokia test
  - Management held accountable at Board level for impediments
  - Training in secret sauce for hyperproductive teams

openview venture partners

How we invented Scrum:

Alan Kay's innovation strategy at Xerox Parc

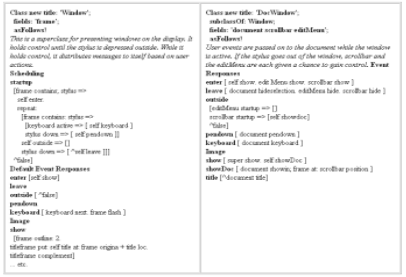
Personal Workstation

Mouse (SRI)

Ethernet

Windows Interface

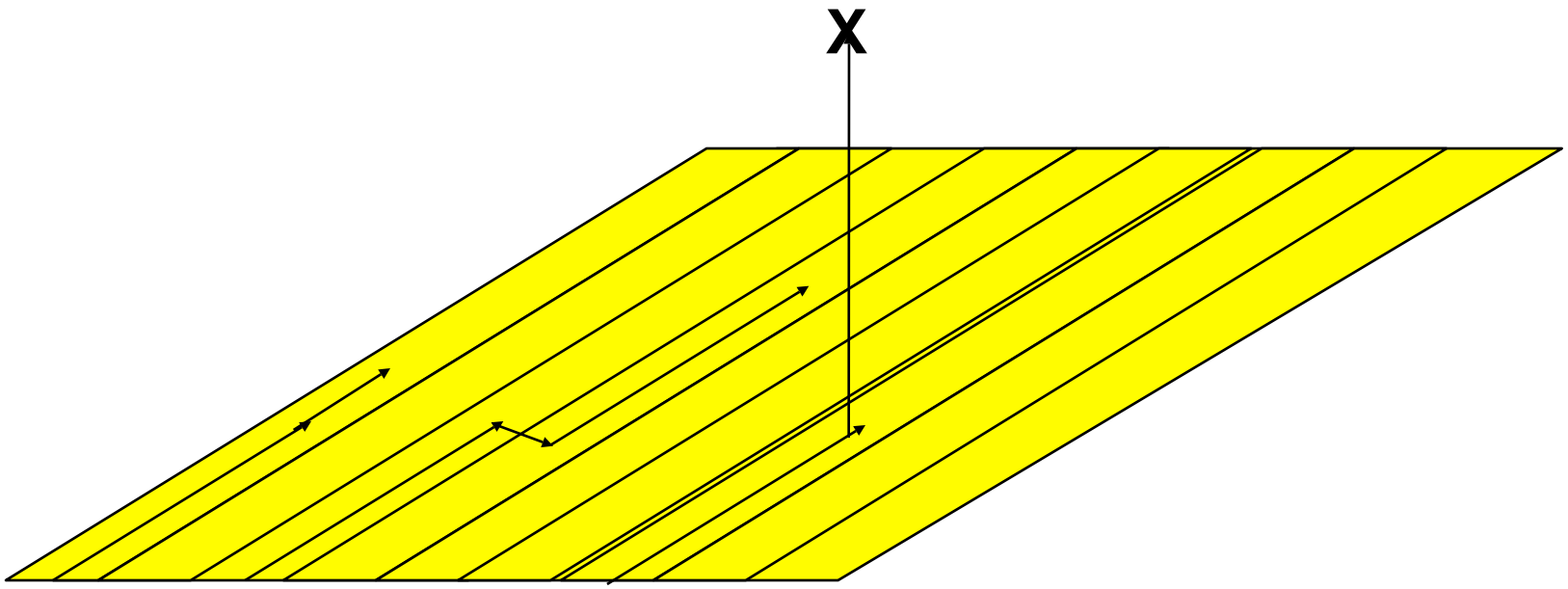
Laser Printer

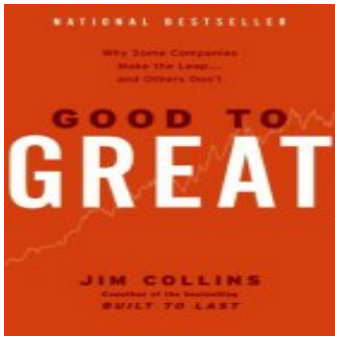


Smalltalk

Alan Kay's Innovation Strategy

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

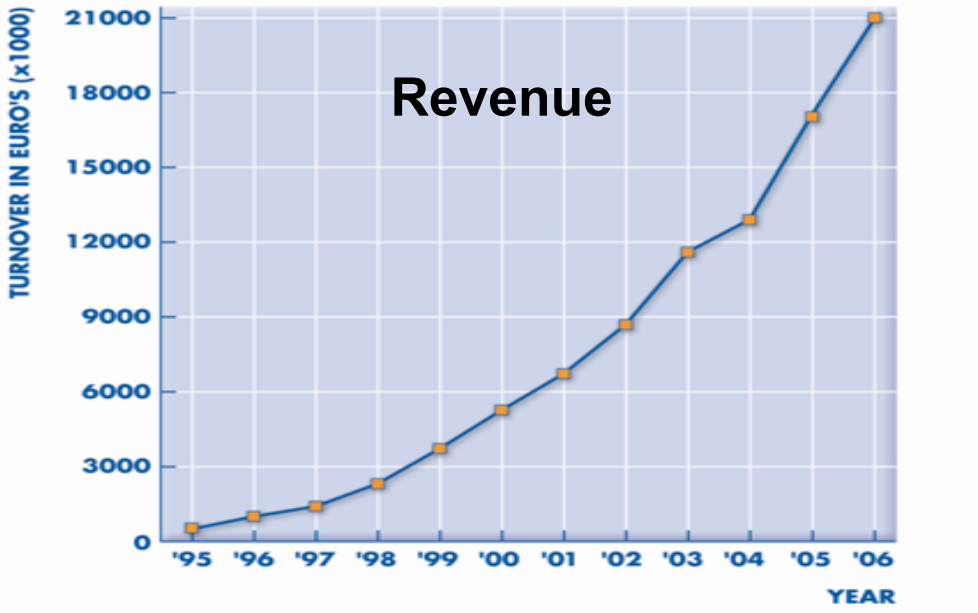


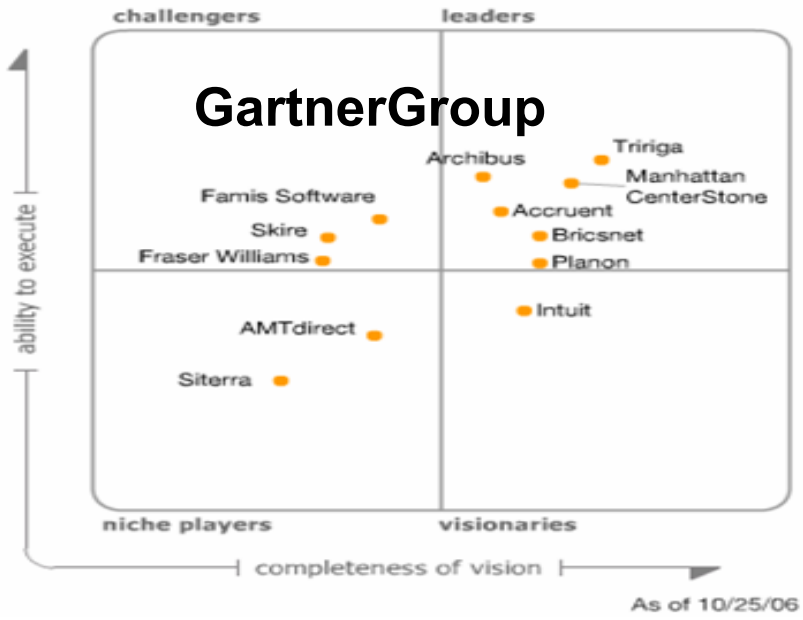


Out of the Box

- 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.

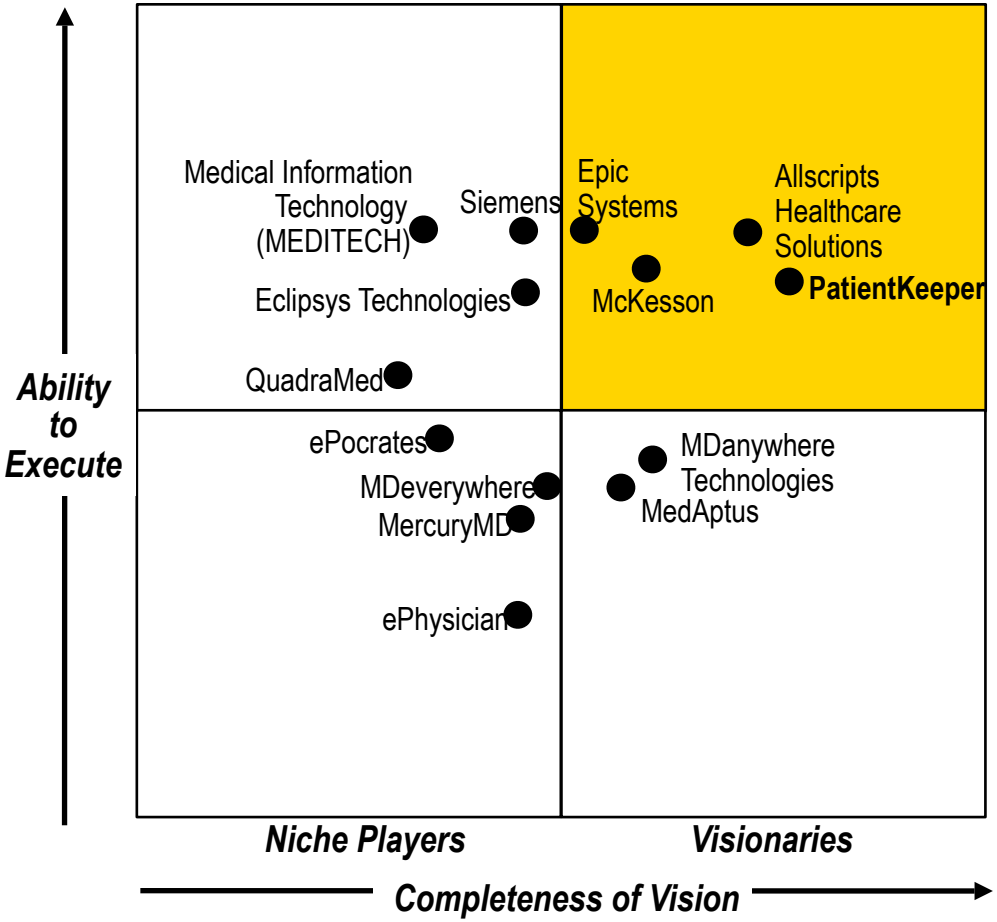
Planon Scrum





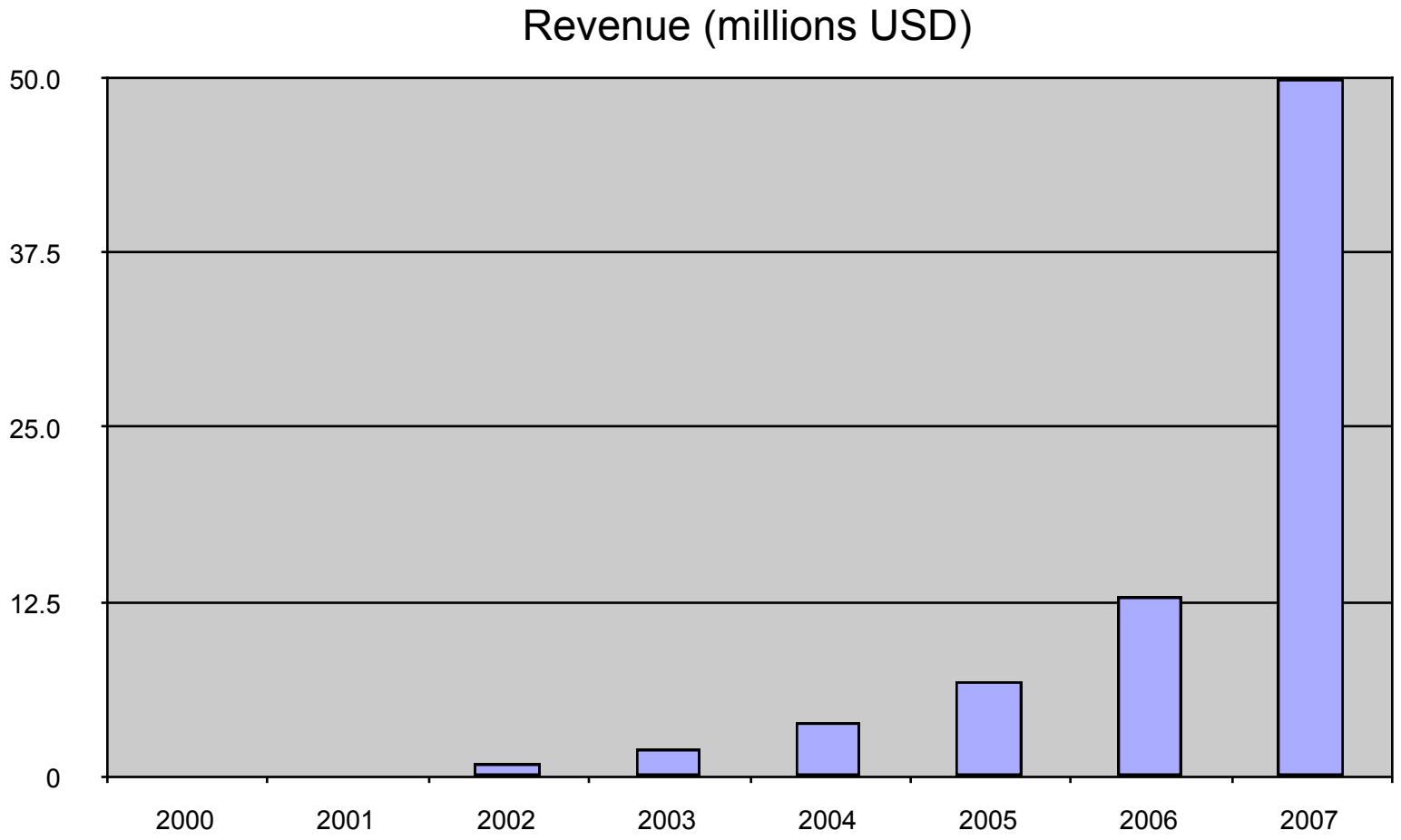
- Great means you are the industry leader in your market and revenue is skyrocketing
- Anyone can aspire to be great!
- That aspiration will make you better

PatientKeeper All-at-Once Scrum



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



ScrumButt Checklist

The Nokia Test by Bas Vodde

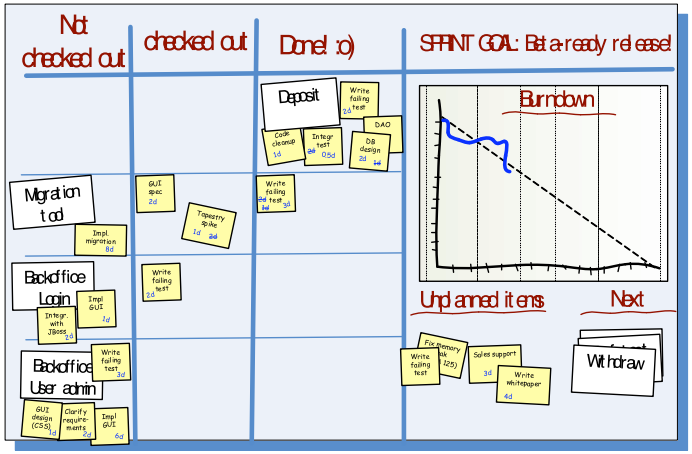
- Are you doing iterative development?
  - Sprints must be time boxed to four weeks or less
  - Software features must be tested and working at the end of an iteration
  - Sprints must start with an Agile specification
- Only 50% of Scrum teams worldwide meet this criteria

http://www.slashphone.com/media/87/7129.html

Are you doing Scrum?

The Nokia Test by Bas Vodde

- Do you know who the product owner is?
- Is there a product backlog prioritized by business value that has estimates created by the team?
- Does the team generate burndown charts and know their velocity?
- You cannot have project managers (or anyone else) disrupting the work of the team.
- Only 10% of teams worldwide meet this criteria.

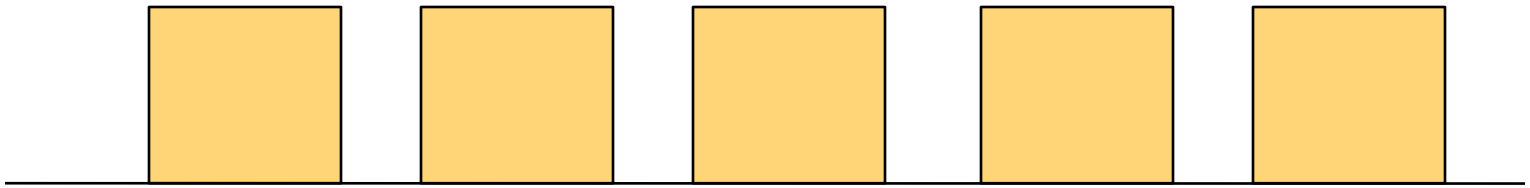


Kniberg, Henrik. Scrum and XP from the Trenches: How We Do Scrum. Version 2.1, Crisp, 5 Apr 2007.

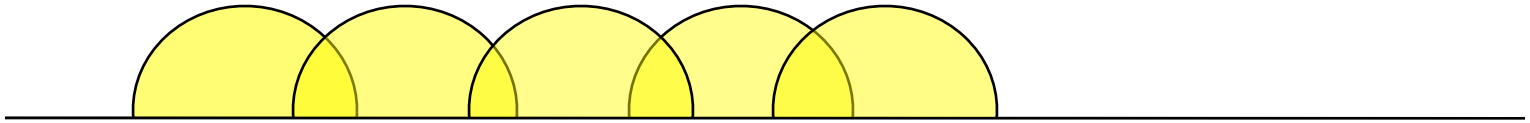
Case Study: Scrum and XP

- 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

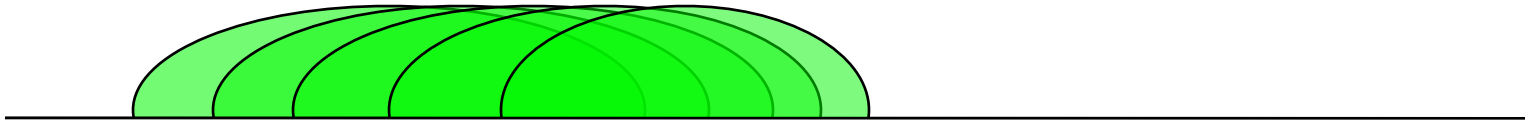
Distributed/Outsourcing Styles



Isolated Scrums



Distributed Scrum of Scrums



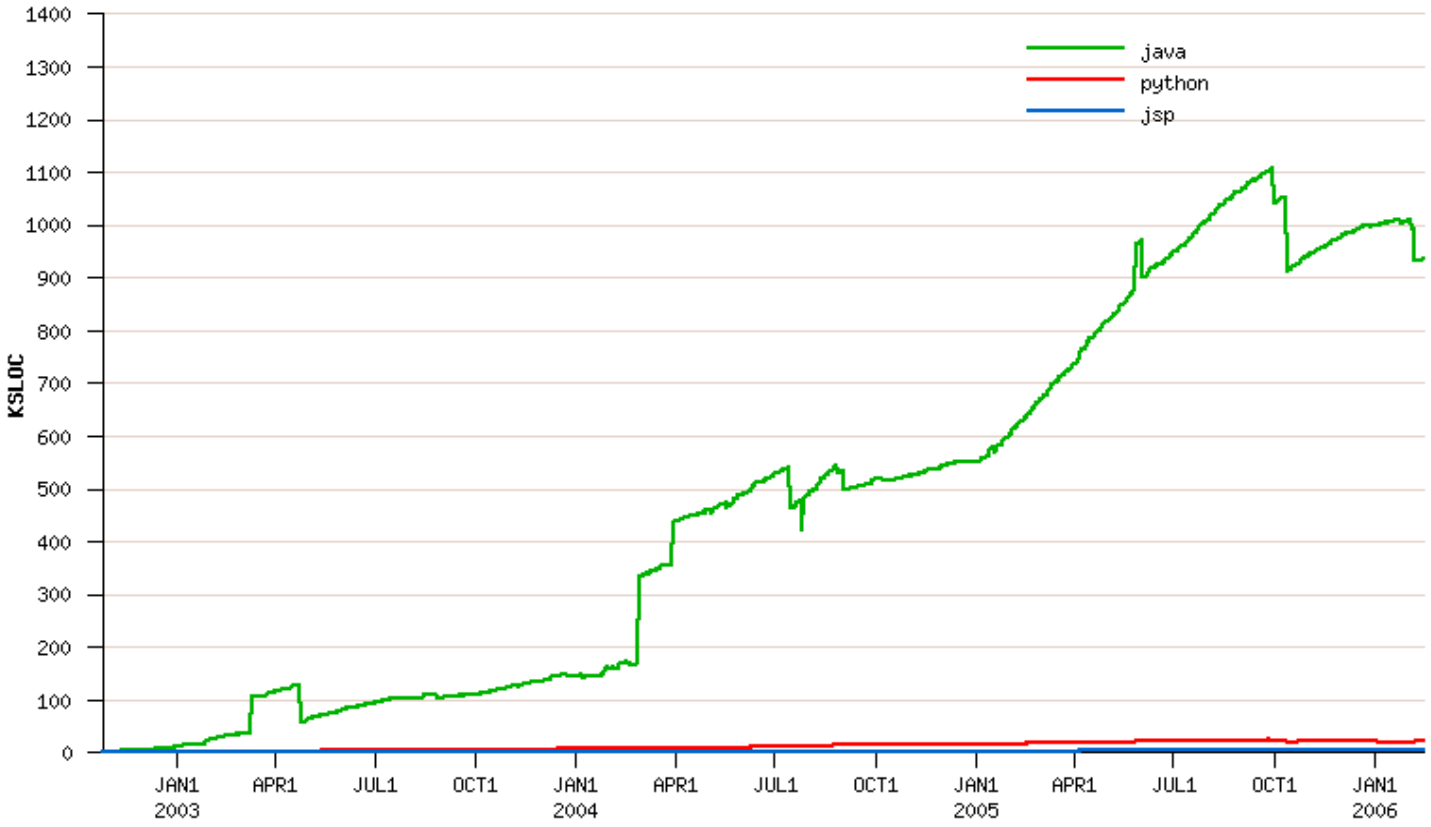
Totally Integrated Scrums

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.

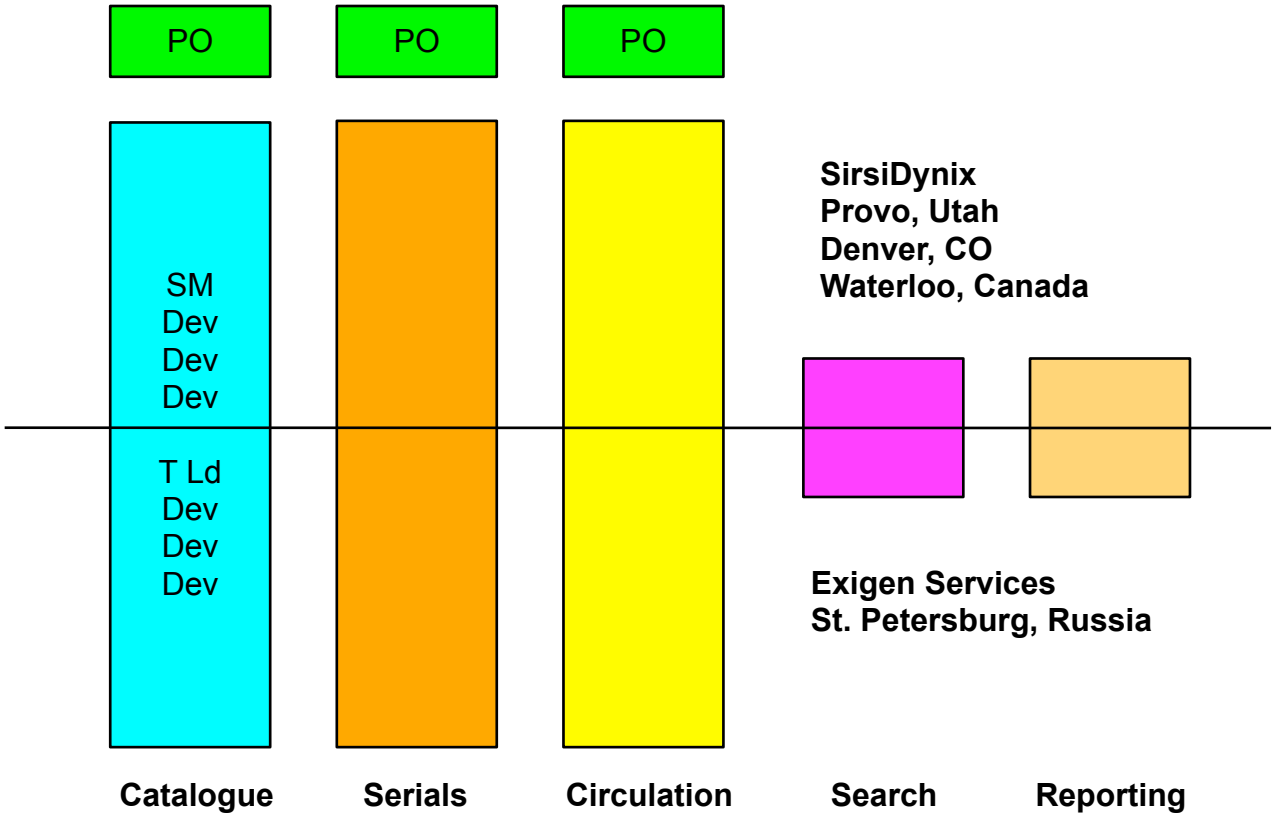
SirsiDynix - Anatomy of a failed project

- Over a million lines of Java code



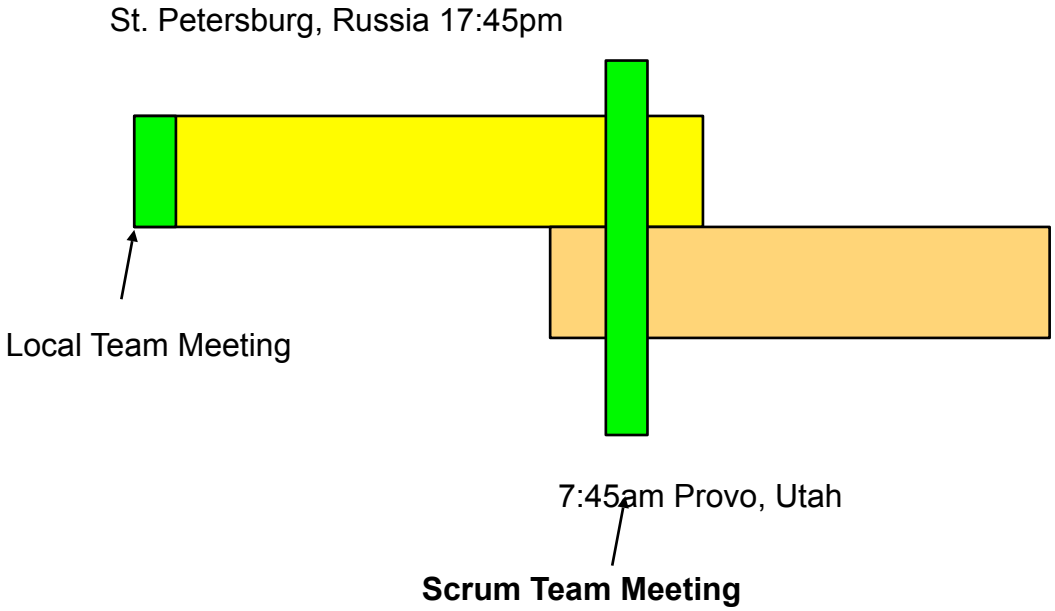
SirsiDynix Distributed Scrum

- 56 developers distributed across sites

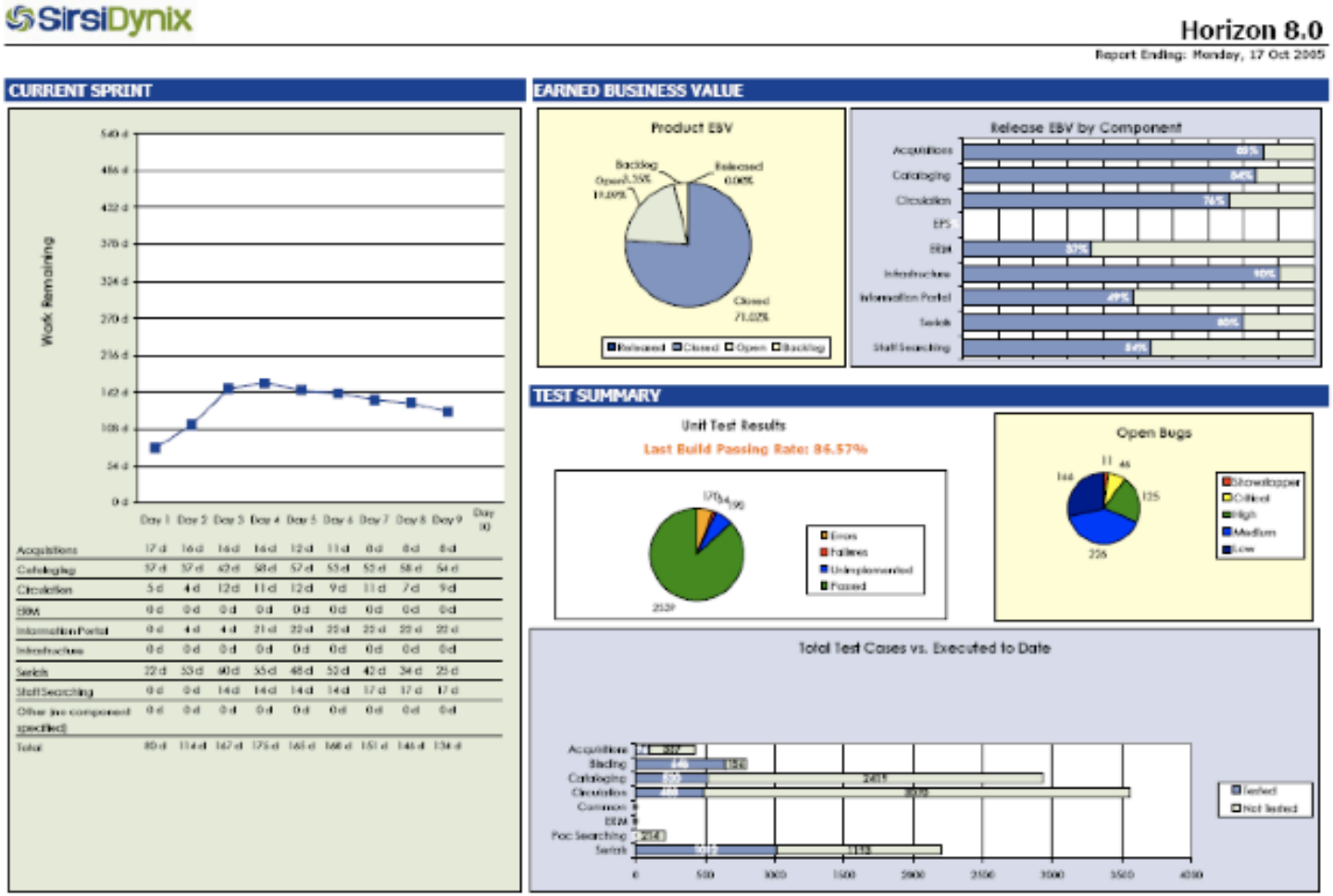


SirsiDynix Distributed Scrum

- Scrum daily meetings



SirsiDynix Distributed Scrum



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,

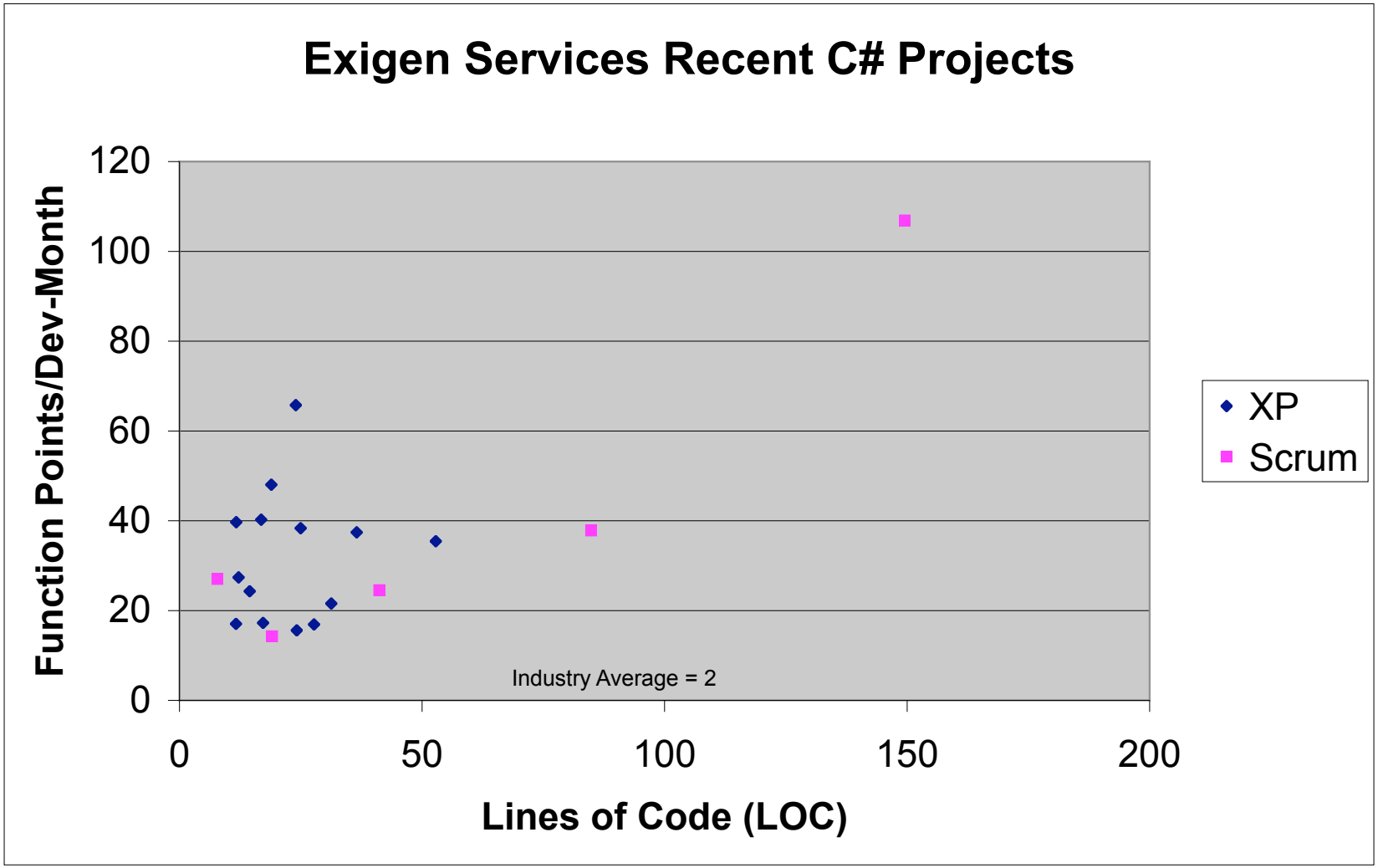
SirsiDynix Challenges

- 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
- 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?

Russian projects velocity data suggests high velocity is not an accident



Simple Minded Venture Capitalist Point of View

- If you can't go 10 times faster than the competition it doesn't 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 not make it go faster.

SirsiDynix Opportunity

- A prospective study should be done
  - full XP technical practices
  - multiple projects
  - meet or exceed SirsiDynix
  - 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.

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.*

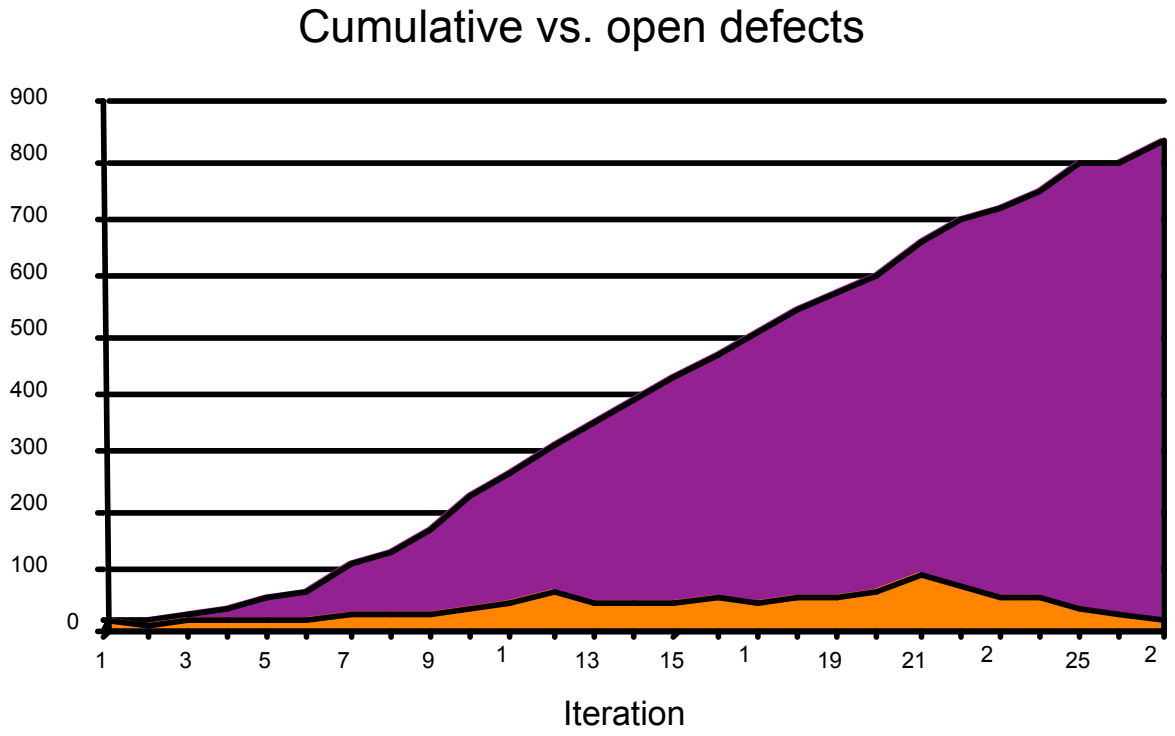
ProRail PUB Example

- 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
- 90% of defects found in an 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.

Agile 2008: Toronto, Aug 4-8

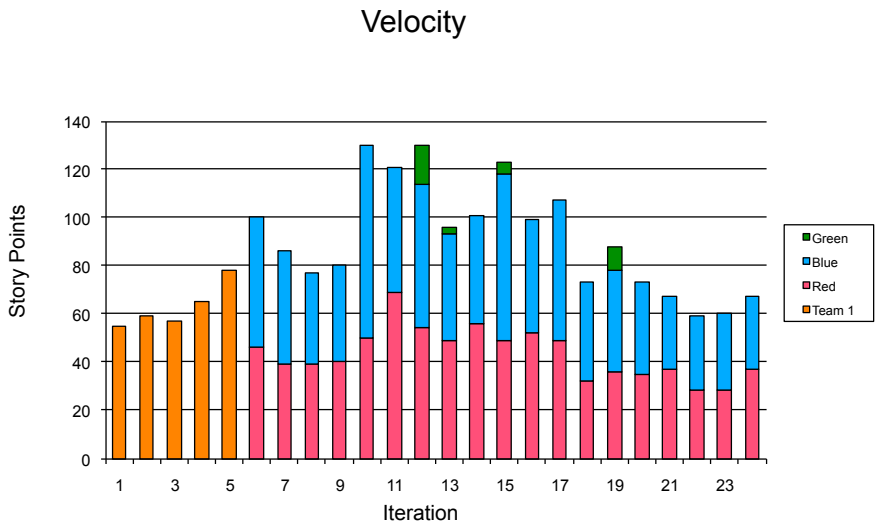
Fully Distributed Scrum: The Secret Sauce for Hyperproductive Outsourced Development Teams

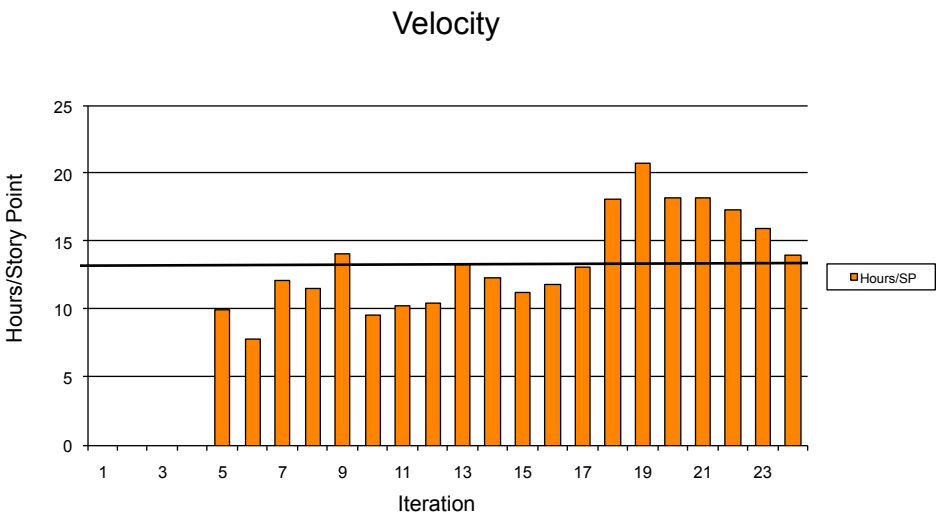
Jeff Sutherland, Ph.D.
Scrum, Inc.
Somerville, MA, USA
jeff.sutherland@computer.org

Guido Schoonheim
Xebia b.v.
Hilversum, Netherlands
gschoonheim@xebia.com

Eelco Rustenburg
Xebia b.v.
Hilversum, Netherlands
erustenburg@xebia.com

Maurits Rijk
Xebia b.v.
Hilversum, Netherlands
mrijk@xebia.com





This graph on the left shows that during the first 6 iterations (data for iteration 6 is missing) the project was executed with one local team (Netherlands). Starting with iteration 7 two project teams were create, each half in India and half in the Nettherlands. The Green Team is a few specialized resources added in the Netherlands at key points in the project. The right graph shows that hous per story point remained relatively stable over time after distribution. The project is a large government project in the Netherlands and full details of the project will be available at Agile 2008.

Dutch Velocity vs. Russian Velocity

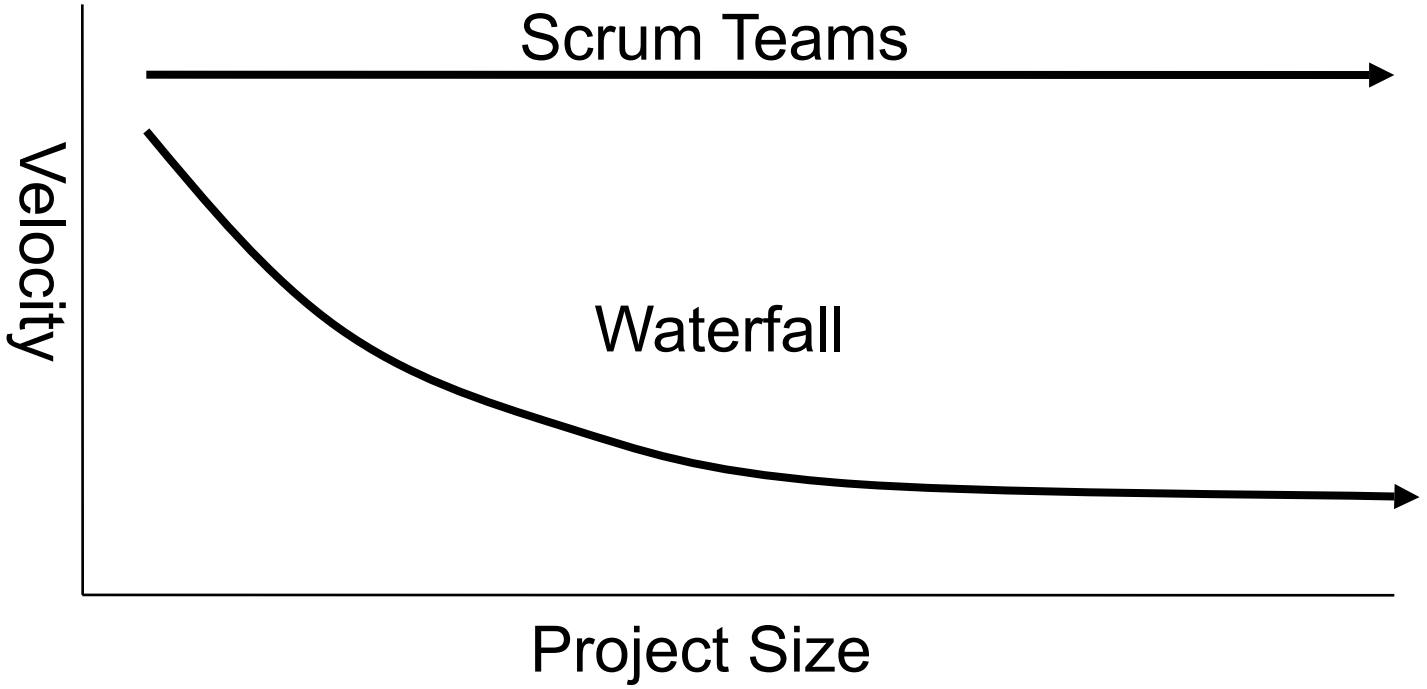
	SirsiDynix[2]	**Xebia[3]**
Person Months	827	125
Lines of Java	671,688	**100,000**
Function Points	12673	**1887**
Function Points per Dev/ Mon	15.3	**15.1**

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)

Comparison of Agile and CMM Results for an Application of 1000 Function Points

	Agile	CMM Level 3	Difference
Size in Function Points	1,000	1,000	0
Size in Java Code Statements	50,000	50,000	0
Monthly burdened cost	$7,500	$7,500	0
Work hours per month	132	132	0
Project staff	5	7	2
Project effort (months)	66	115	49
Project effort (hours)	8,712	15,180	6,486
Project schedule (months)	14	19	5
Project cost	$495,000	$862,500	$367,500
Function Points per Month	**15.15**	**8.67**	**-6.46**
Work hours per function point	8.71	15.18	6.47
LOC per month	758	435	-323
Function point assignment scope	200	143	-57
LOC assignment scope	10,000	7,143	-2,857
Cost per function point	$495	$863	$368
Cost per LOC	$9.90	$17.25	$7.35
Defect potential	4,250	4,500	250
Defect potential per function point	4.25	4.50	0.25
Defect removal efficiency	90%	95%	0.5%
Delivered defects	425	225	-200
High-severity defects	128	68	-60

Linear Scalability of Large Scrum Projects

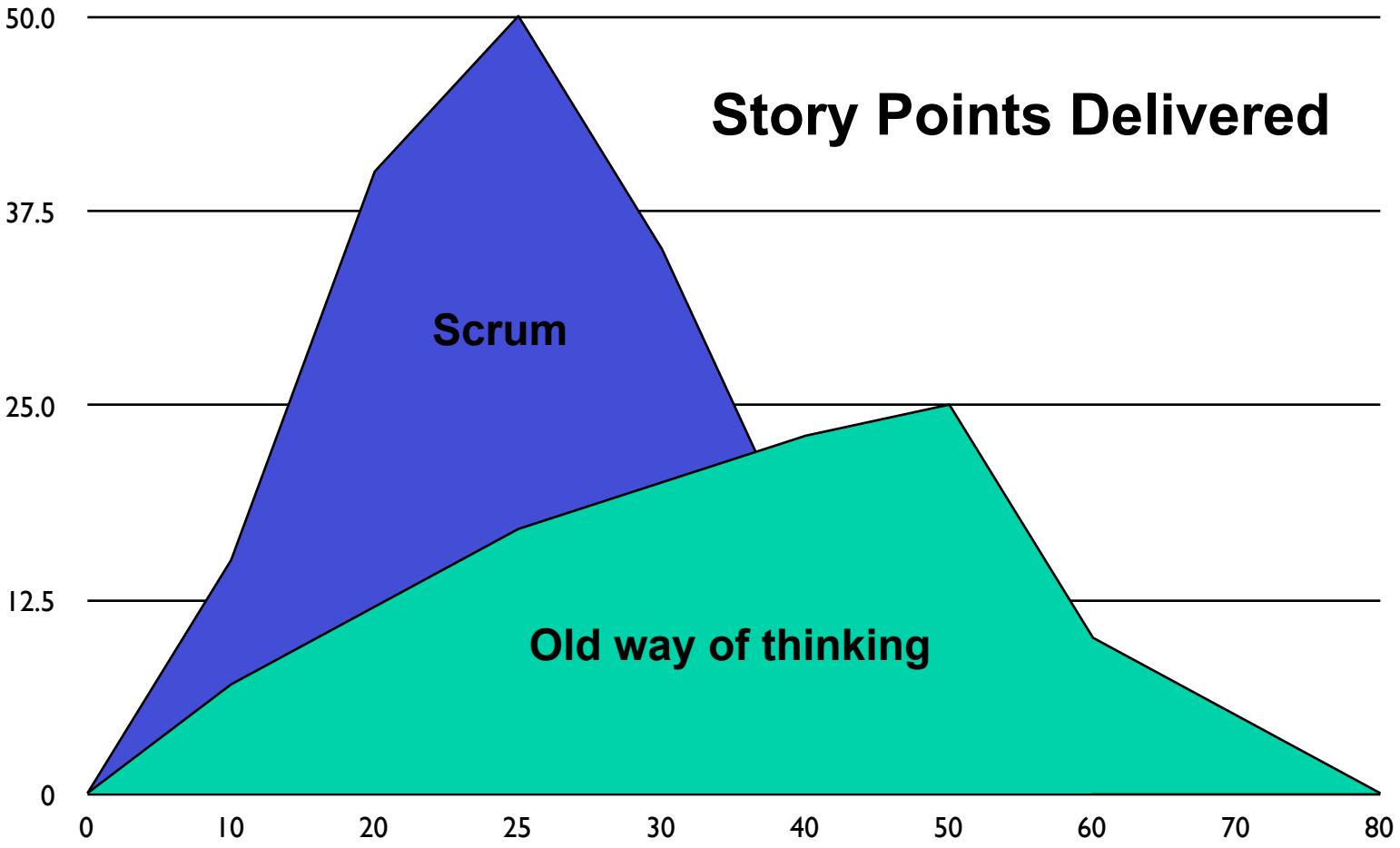


•J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii, 2007.
•J. Sutherland, C. Jacobson, and K. Johnson, "Scrum and CMMI Level 5: A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

SirsiDynix vs. Xebia

- SirsiDynix issues
  - SirsiDynix first implementation of fully distributed model
  - No administrative costs in function point data
  - Quality issues
    - working builds only at Sprint boundaries
    - test cases not run daily
    - lack of refactoring caused major rework late in project
    - no test driven development
- Xebia OneTeam
  - Fully distributed team model replicated across multiple projects
  - Fully burdened function point data including administrative costs
  - Continuous integration of working builds with full unit tests
  - Constant refactoring of code included in definition of done
  - Fully automated daily reqression and performance testing
  - Acceptance test of full system by customer every two weeks with virtually no defects

Maxwell Curve



www.openviewpartners.com

Recommendations

- For those who can execute “Pretty Good Scrum”, why not go from good to great?
- Using fully distributed teams requires initial colocation, careful measurement of velocity, and clear definition of done.
- Breakdown of team formation can be detected by measuring cost per story point.
- Increasing cost can mean one or more of the following:
  - breakdown of team formation
  - loss of quality causing increased rework
- Those who measure their results can correct their problems. Those who don’t suffer eternal dysfunction.

Questions?

Secret Sauce #1

- Get your teams to pass the Nokia test.

Secret Sauce #2

- Get management totally involved.

Secret Sauce #3

- Use communication saturation as a competitive advantage.
- What backlog item will drive the earliest appearance of a high priority feature that can be tested?
- Entire team decides this and it is the first thing selected to be worked on.

Secret Sauce #4

- Extend the definition of done

Best practices for OpenView portfolio companies

- Establish a stable Scrum environment
  - ensure Nokia test is implemented
  - get management totally involved
  - demonstrate 100% gain in velocity
  - should be easily achievable in 6 months
    - management support in removing impediments required
    - management training in Agile development and lean production essential

Secret Sauce for Hyperproductive State

- Team passes Nokia test
- ScrumMaster removes impediments in priority order
- Management is totally involved
  - assists ScrumMaster in removing impediments
  - Strong business imperative
  - Crystal clear product backlog by product owner
- Optimize Maxwell curve
- Team maximizes communication saturation
- Extend definition of Done
- Agressively execute Scrum emergency procedure