



AGILE CONTRACTS

Money for Nothing and Your Change for Free

With help from Citrix Online, Google, Yahoo, Microsoft, IBM, Oracle, MySpace, Adobe, GE, Siemens, Disney Animation, BellSouth, Alcatel-Lucent, GSI Commerce, Ulticom, Palm, St. Jude Medical, DigiChart, RosettaStone, Healthwise, Sony/Ericsson, Accenture, Trifork, Systematic Software Engineering, Exigen Services, SirsiDynix, Softhouse, Philips, Barclays Global Investors, Constant Contact, Wellogic, Inova Solutions, Medco, Saxo Bank, Xebia, Insight.com, SolutionsIQ, Crisp, Johns Hopkins Applied Physics Laboratory, Unitarian Universalist Association, Motley Fool, Planon, FinnTech, 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 Hamilton, Scrum Alliance, Fortis, DIPS, Program UtVikling, Sulake, TietoEnator, Gilb.com, WebGuide Partner, Emergn, NSB (Norwegian Railway), Danske Bank, Pegasystems, Wake Forest University, The Economist, iContact, Avaya, Kanban Marketing, accelare, Tam Tam, Telefonica/O2, iSense



Jeff Sutherland, Ph.D.



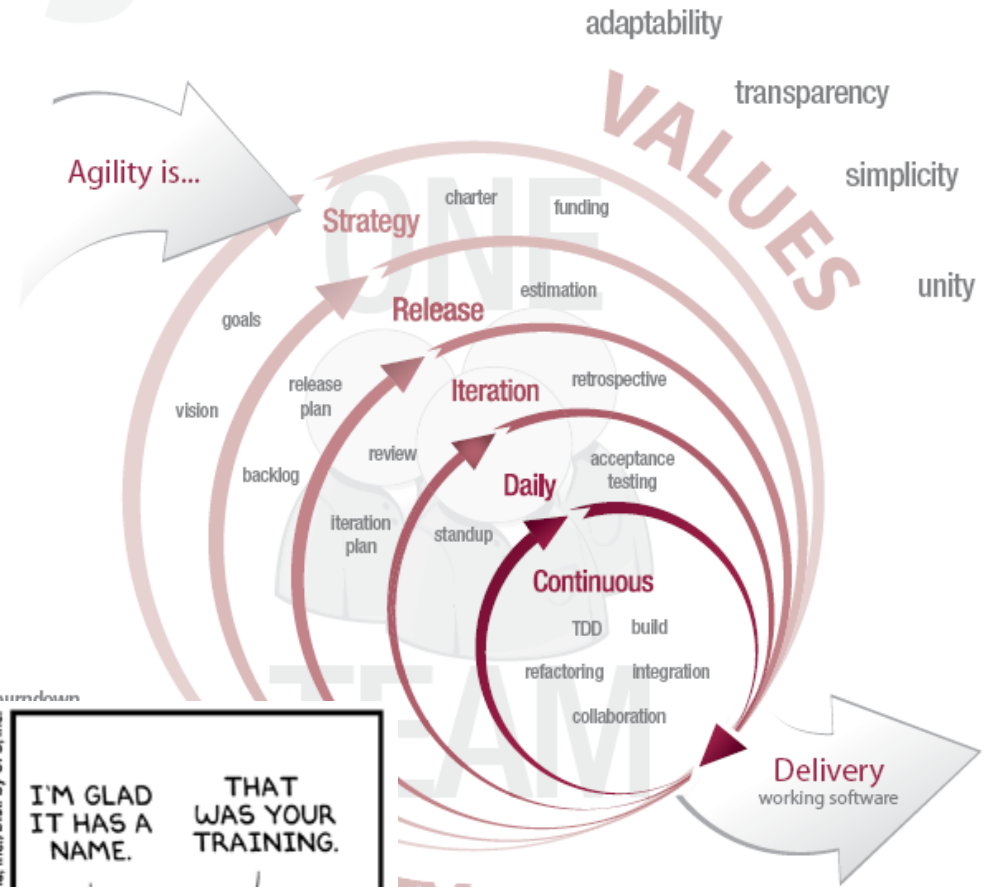
- **Chairman, Scrum Training Institute**
- **CEO Scrum, Inc. and Senior Advisor, OpenView Venture Partners**
 - Agile coach for OpenView Venture Partners portfolio companies
 - CEO/CTO/VP Engineering for 11 software companies
 - Created first Scrum at Easel Corp. in 1993. Rolled out Scrum in next 7 companies
 - Achieved hyperproductive state in all companies. Signatory of Agile Manifesto and founder of Agile Alliance
- <http://scrum.jeffsutherland.com>
- jeff@scruminc.com



© Jeff Sutherland 1993-2010

Agile Development

ScrumButt



Accelerate Success

Avoiding ScrumButt

Nokia Test Origins







- **In 2005, Bas Vodde started training and coaching teams at Nokia Networks in Finland. The first Nokia test focused on Agile practices**
 - jeffsutherland.com/basvodde2006_nokia_agile.pdf
- **By 2007, Siemens joined 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
 - jeffsutherland.com/Agile2008MoneyforNothing.pdf

Nokia Test covers these areas:

- Iterations
- Testing
- Enabling Specifications
- Product Owner
- Product Backlog
- Estimates
- Burndown
- Disruption
- Team










Question 1 - Iterations

-  **No iterations - 0**
-  **Iterations > 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 testers on team - 0**
- 🕒 **Unit tested - 1**
- 🕒 **Features tested - 5**
- 🕒 **Features tested as soon as completed - 7**
- 🕒 **Software passes acceptance testing - 8**
- 🕒 **Software is deployed - 10**

Question 3 - Enabling Specifications

-  **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 has a 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 has good user stories that satisfy the INVEST criteria - 5**
- 🕒 **Two sprints of Product Backlog are in a READY state - 7**
- 🕒 **Product Roadmap is available and updated regularly based on team estimates of Product Backlog - 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 - Sprint 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 is done (TrackDone pattern) - 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**

ScrumPloP pattern: Track Done by Jim Coplien

It is easy to interpret the burn-down chart as a good portrayal of estimated remaining time, and to use that portrayal to develop confidence in meeting the Sprint's actual business goals of done functionality.

Usually, team members update the burn-down chart daily to reflect adjustments to the amount of remaining work. Such updates reflect a desire to have as good knowledge as is possible about the effort remaining. These estimates are made in mid-stream and reflect increases that arise from emergent requirements. However, given that one emergent requirement has been discovered in a task doesn't imply that no others remain. While the confidence in an estimate usually improves with each revision and with continued work on the task, unusually wicked problems seem never to converge.

On the other hand the Product Owner is not centrally interested in partially completed work, only in items that are done and potentially shippable. Since the goal of Scrum is to achieve the Sprint target agreed with the Product Owner, and to reduce risk, the focus should be on done. Emergent requirements increase risk, and the Product Owner is certainly interested if estimates expand. Because there may always be emergent requirements, any estimate of remaining time based on work mid-stream in a task has a higher degree of uncertainty than the relatively risk-free estimate of zero remaining time for done items.

In theory, it is possible for the remaining time on a burn-down chart to be quite near zero, yet to have few (or perhaps zero!) tasks in the done state.

Therefore:

Update the Product Backlog in only two cases: reducing the amount of remaining known work if the task is done; and increasing the amount of known work if the task grows in size due to emergent requirements or other insights gained during the Sprint. Do not reduce the amount of remaining work that arises from progress on partially completed tasks.

* * *

The team and Product Owner have a better picture of the state of the Sprint with respect to the Sprint's business goals of delivering done functionality. The team can revise estimations in the middle of a Sprint with more confidence because they are not dependent on the unknown remaining time for partially completed tasks. Yet, the risks incurred by the "surprises" of emergent requirements are embraced and made visible.

It is impossible, using this approach, to come near the end of a Sprint with a burn-down chart that projects success even if the Sprint only ends with 90% of the tasks 90% done.

* * *

There is a chance that a completed task can become "un-completed" by emerging requirements in some other task during the sprint. For such cases, see the pattern Domino Effect.

This pattern was suggested by Jeff Sutherland, co-founder of Scrum, and he reports that it is widely used by his clients.

James O. Coplien
Wednesday, September 24, 2008

© Jeff Sutherland 1993-2010

Question 8 - Team Disruption

- 🌐 **Manager or Project Leader disrupts team - 0**
- 🌐 **Product Owner disrupts team - 1**
- 🌐 **Managers, Project Leaders or Team leaders telling people what to do - 3**
- 🌐 **Have Project Leader and Scrum roles - 5**
- 🌐 **No one disrupting team, only Scrum roles - 10**

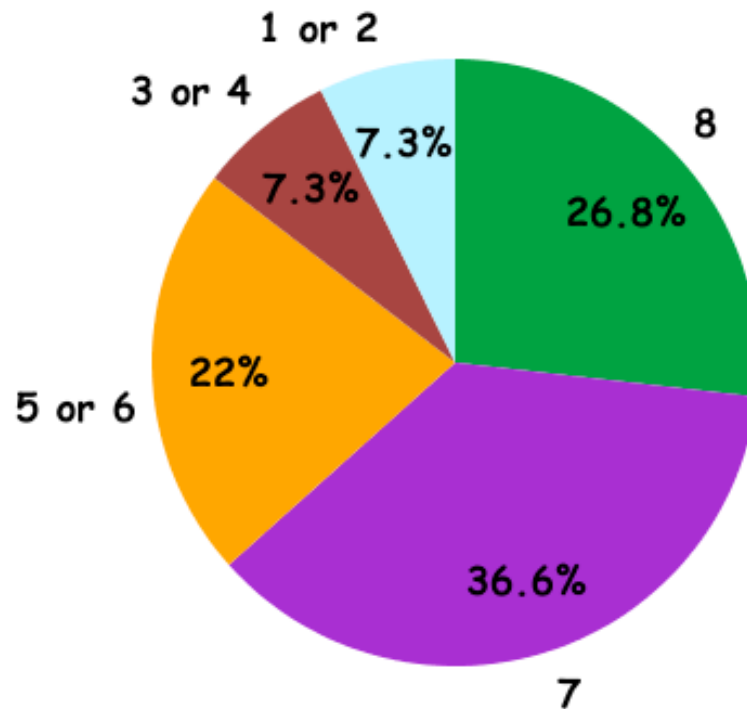
Question 9 - Team

- 🌐 **Tasks assigned to individuals during Sprint Planning – 0**
- 🌐 **Team members do not have any overlap in their area of expertise – 0**
- 🌐 **No emergent leadership - one or more team members designated as a directive authority -1**
- 🌐 **Team does not have the necessary competency - 2**
- 🌐 **Team commits collectively to Sprint goal and backlog - 7**
- 🌐 **Team members collectively fight impediments**

ScrumButt

74% of Scrum teams

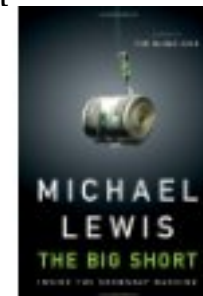
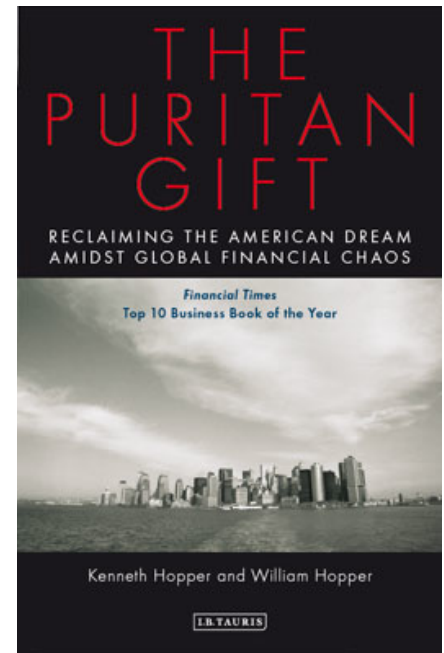
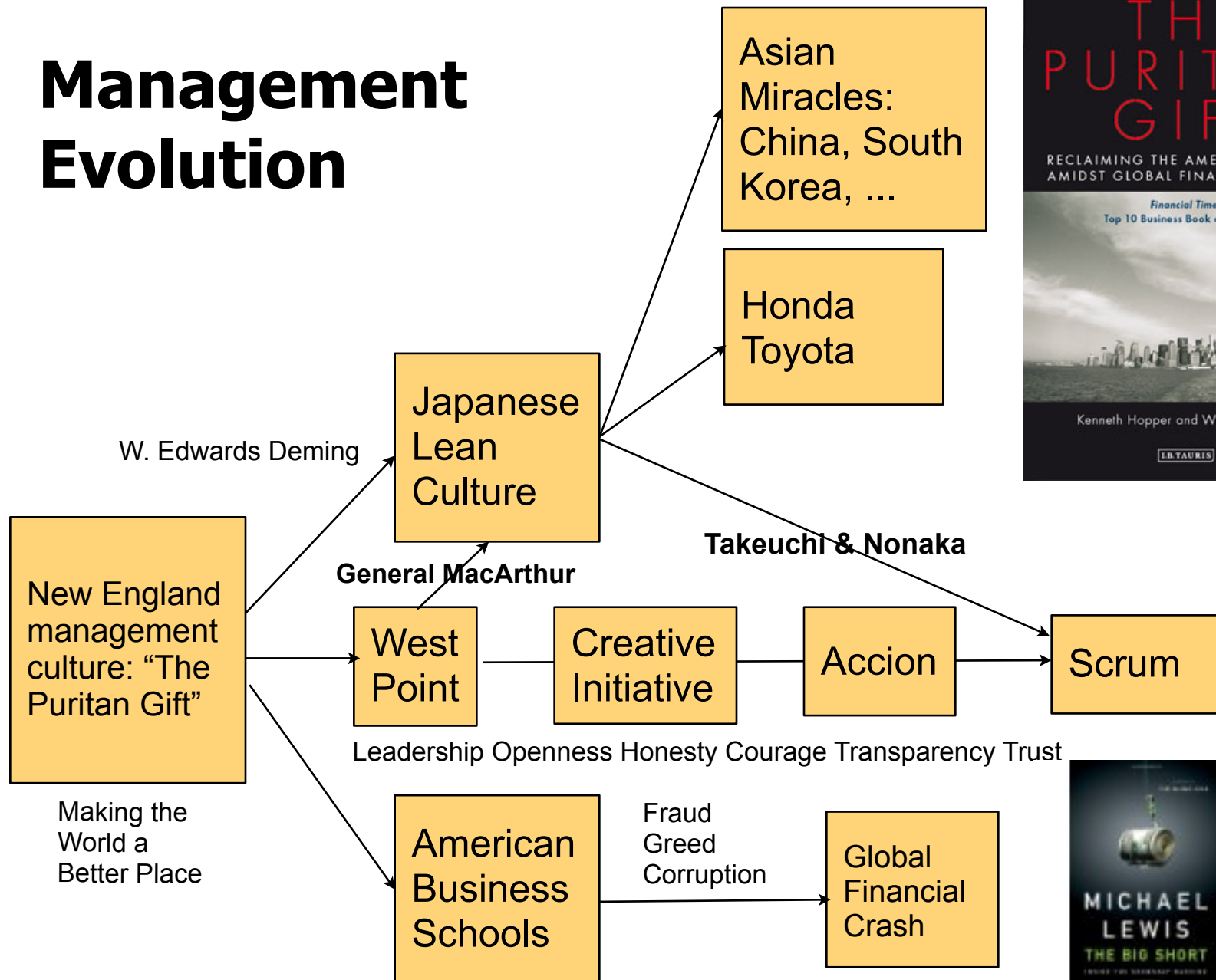
Nokia Test - Overall Scores



(c) 2008 Peter Stevens
scrum-breakfast.com

© Jeff Sutherland 1993-2010

Management Evolution



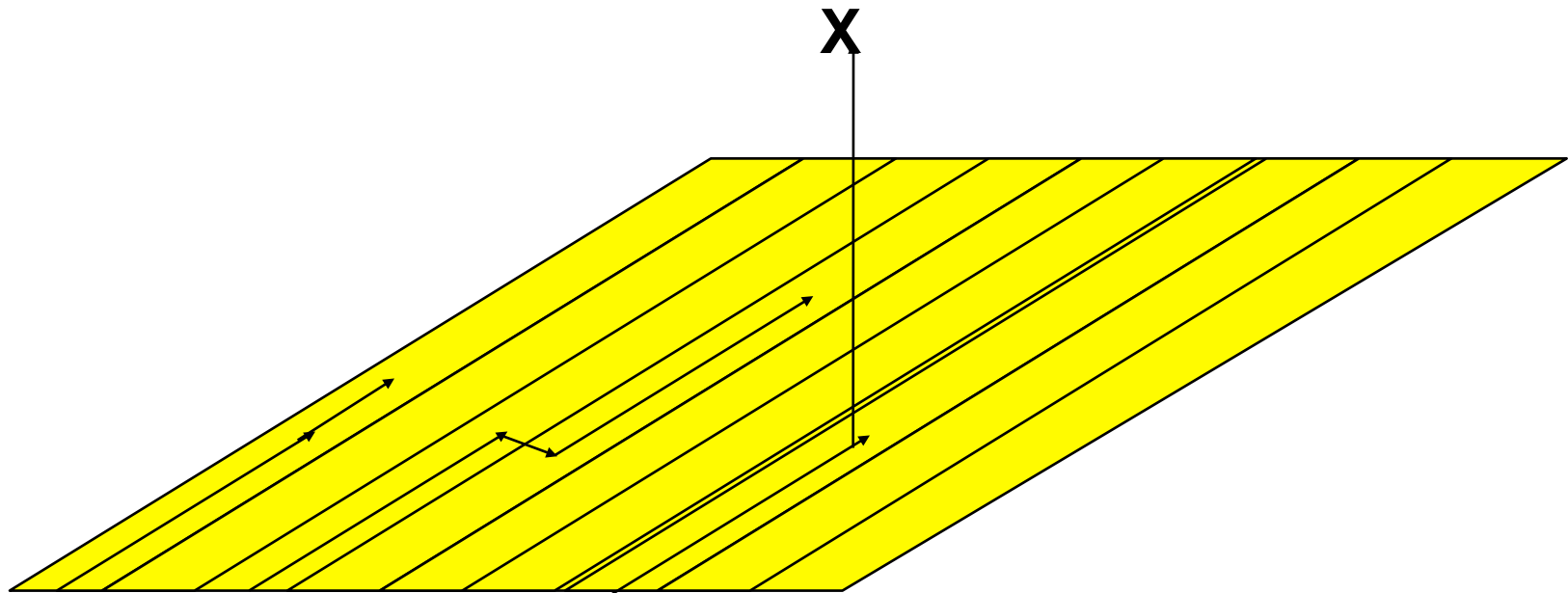
© Jeff Sutherland 1993-2010



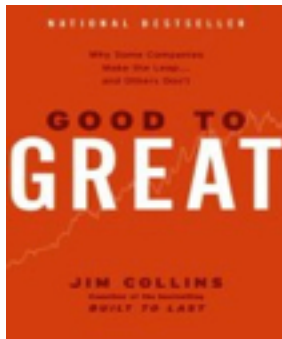
Smalltalk

Alan Kay's Innovation Strategy

- Incremental - NO
- Cross Discipline - NYET
- Extreme data points - YES



© Jeff Sutherland 1993-2010



Benchmarked Out of the Box



Scrum looked at projects off the chart

- (IBM Surgical Team) F. P. Brooks, *The Mythical Man Month: Essays on Software Engineering*: Addison-Wesley, 1995.
- Takeuchi and Nonaka. [The New New Product Development Game](#). Harvard Business Review, 1986
- J. O. Coplien, "Borland Software Craftsmanship: A New Look at Process, Quality and Productivity," in 5th Annual Borland International Conference, Orlando, FL, 1994.



Scrum: A Pattern Language for Hyperproductive Software Development

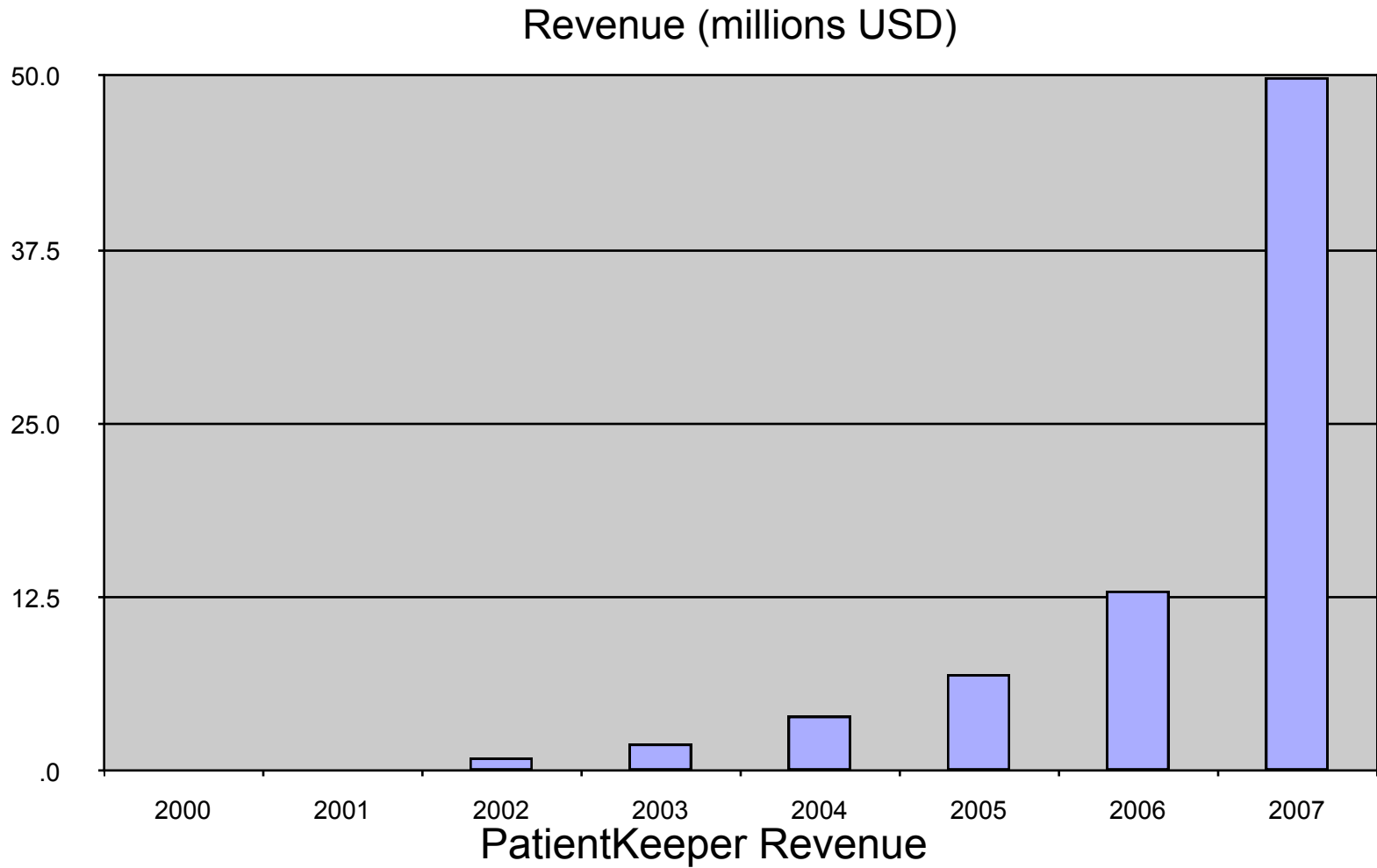
- By M. Beedle, M. Devos, Y. Sharon, K. Schwaber, and J. Sutherland. In *Pattern Languages of Program Design*. vol. 4, N. Harrison, Ed. Boston: Addison-Wesley, 1999, pp. 637-651.



Every team can achieve hyperproductivity

- J. Sutherland, S. Downey, and B. Granvik, "[Shock Therapy: A Bootstrap for a Hyper-Productive Scrum](#)" in *Agile 2009*, Chicago, 2009.
- C. Jakobsen and J. Sutherland, "[Scrum and CMMI – Going from Good to Great: are you ready-ready to be done-done?](#)," in *Agile 2009*, Chicago, 2009.

Pretty Good to Great Scrum



© Jeff Sutherland 1993-2010

Another way to measure ScrumButt

- **Great 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 - average 35% productivity improvement

ScrumButt vs. Scrum Design Goal



© Jeff Sutherland 1993-2010

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 Nokia test
 - Management held accountable at Board level for removing impediments
 - Implementation of hyperproductive Scrum



© Jeff Sutherland 1993-2010

Velocity in Function Points/Dev month

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

© Jeff Sutherland 1993-2007
© Jeff Sutherland 1993-2010

Russian Velocity = Dutch 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)

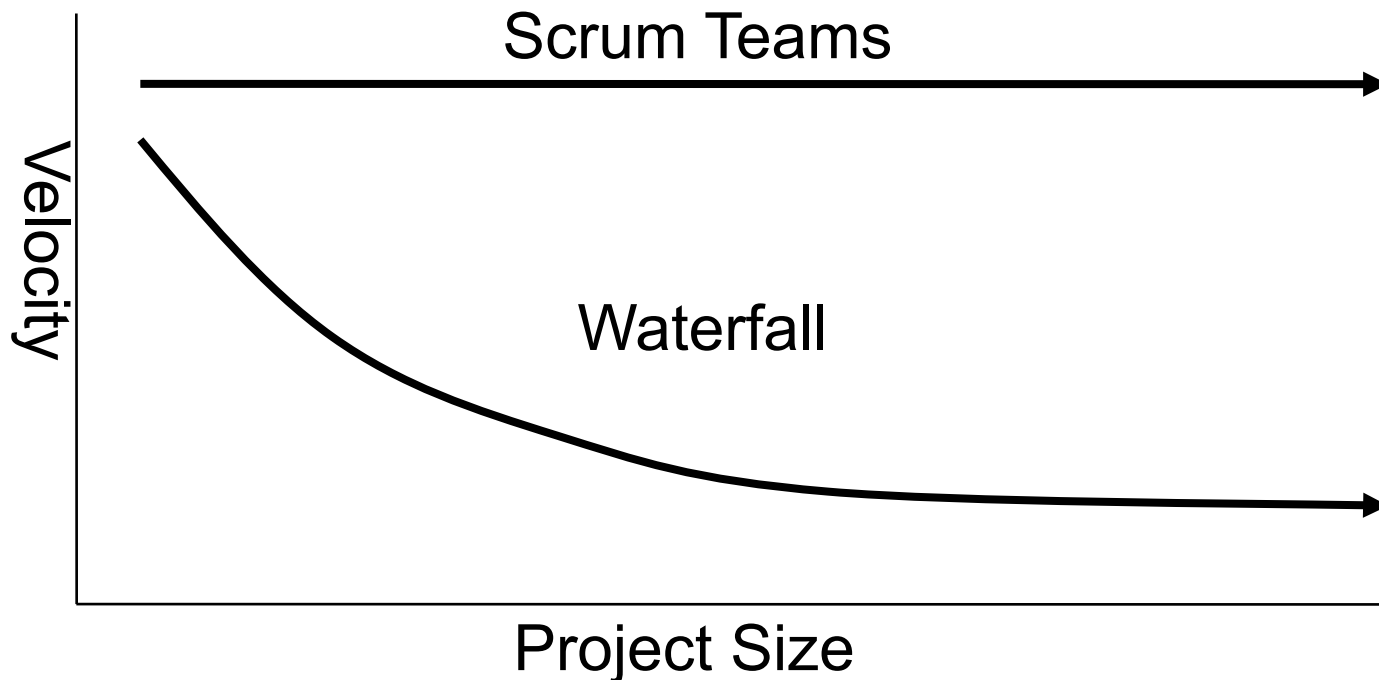
© Jeff Sutherland 1993-2010

Comparison of Agile and CMM Results for an Application of 1000 Function Points - Capers Jones 2008

	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

© Jeff Sutherland 1993-2010

Linear Scalability of Pretty Good 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.

If a company can deliver great Scrum, how can they monetize their performance?

- **Industry incentives now are for projects to be late.**
- **Many vendors only make money if the project is late and over budget due to change requests and building functionality the end users do not want.**
- **CIOs participate in this disfunctional behavior using their current proposal and contracting process.**
- **The whole industry could be viewed as driven by bad incentives and faulty practices as 83% of waterfall projects over \$3M fail - see Gartner Group summary of Standish data.**

Typical Fixed Price Contract

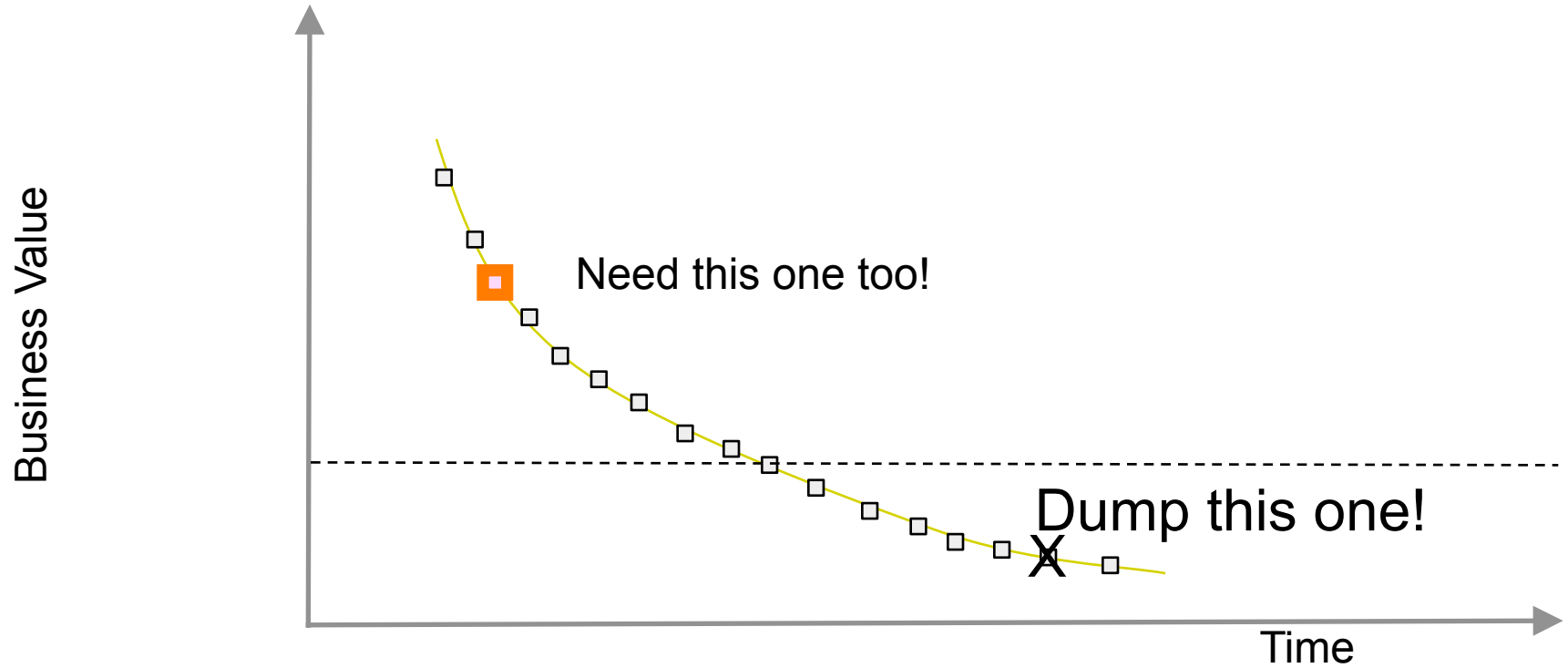
- **Client sends out tender to 3+ potential suppliers. Everything is equally important. Assume total is \$5M.**
- **All suppliers place a bid of around \$5M.**
- **One supplier chosen and contract signed.**
- **Change requests start coming in from day one. All changes are expensive. Project ends up with millions of dollars in change requests.**
- **After acceptance there still are more work to do because of bugs and some functionality that is not really completed or useful.**
- **Project cost at end is \$10M - delivered late.**

The Alternative - Change for Free

- **Use a standard fixed price contract which includes time and materials for changes**
- **Insert the Change for Free option clause.**
 - The customer must execute this option by working with the Scrum Team every Sprint.
 - Failure to do this voids this clause and the contract reverts to time and materials.
- **The Scrum Product Owner reprioritizes the Product Backlog at the end of each Sprint.**
- **Changes are included with these rules**
 - Changes in priorities are free if total contract work is not changed
 - New features may be added for free at Sprint boundaries if low priority items of equal work are removed from contract.
- **Requirements of customer:**
 - Features are prioritized by business value and implemented in order of maximum value
 - Users follows project closely and work with the Product Owner to produce a quality Product Backlog

© Jeff Sutherland 1993-2010

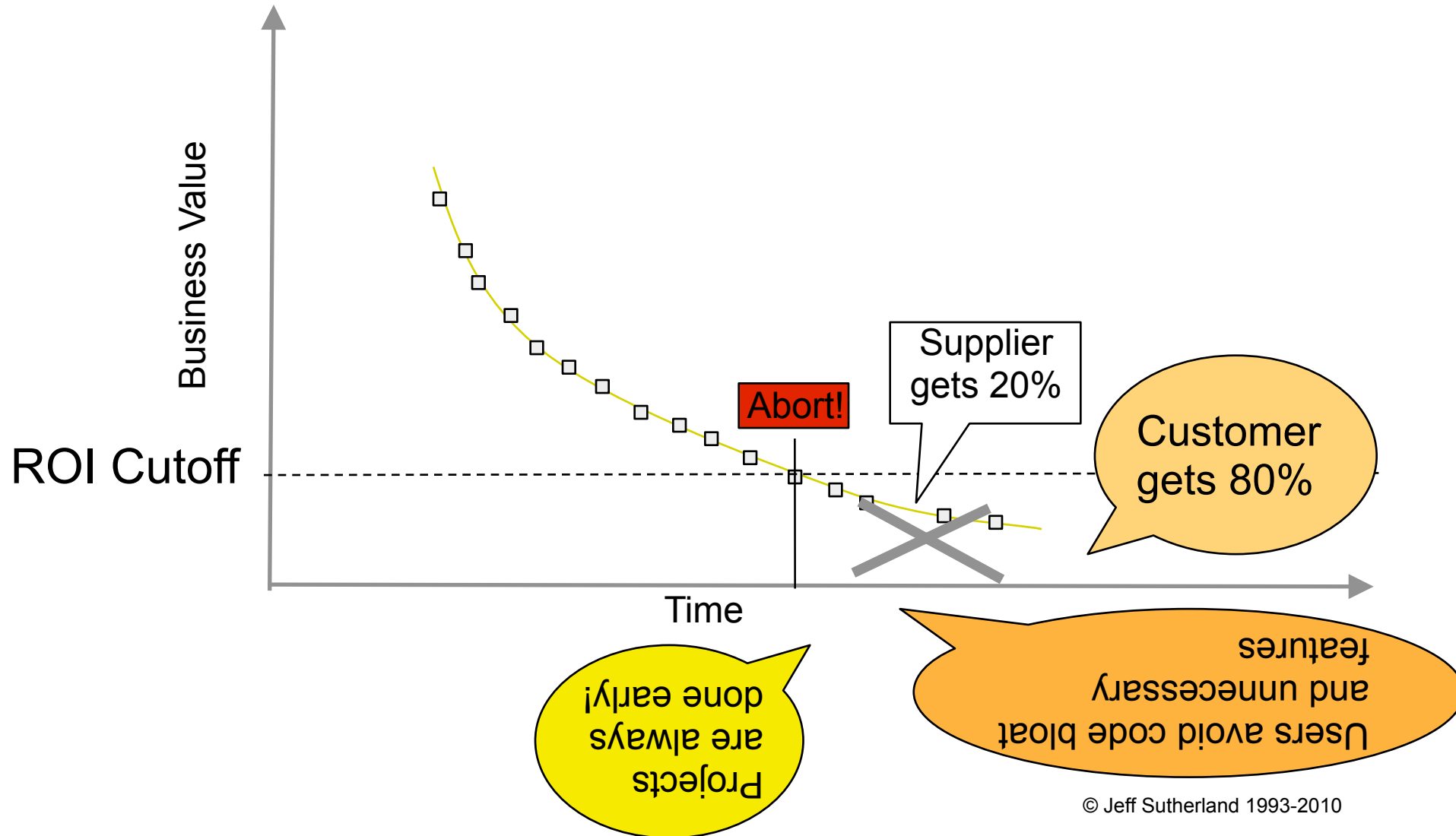
Change for free!



We can do better than Change for Free Money for Nothing!

- **Use standard fixed price contract**
- **Insert Money for Nothing clause.**
 - Only operational if customer follows Scrum rules
 - Mutually agreed estimates for all work items
 - Otherwise contract reverts to time and materials
- **Customer determines ROI cutoff where implementation of the next feature costs more than the value of the feature.**
- **Supplier allows termination of contract at any time for 20% of remaining contract value.**
- **Supplier assumes risk of late delivery of mutually**

Money for Nothing!



© Jeff Sutherland 1993-2010

Fixed Price, Fixed Date

Money for Nothing and Change for Free

Contract provisions:

1. **Customer involvement allows us to tune the system to the latest known business value.**
2. **Any requirement that hasn't already been worked on can be swapped out for another of equal value;**
3. **Priority of requirements can be changed by customer;**
4. **Customer may request additional releases at any time at prevailing time and material fees;**
5. **Customer may terminate contract early if value has been satisfied for 20% of remaining unbilled contract value**

Fixed Resources, Fixed Date

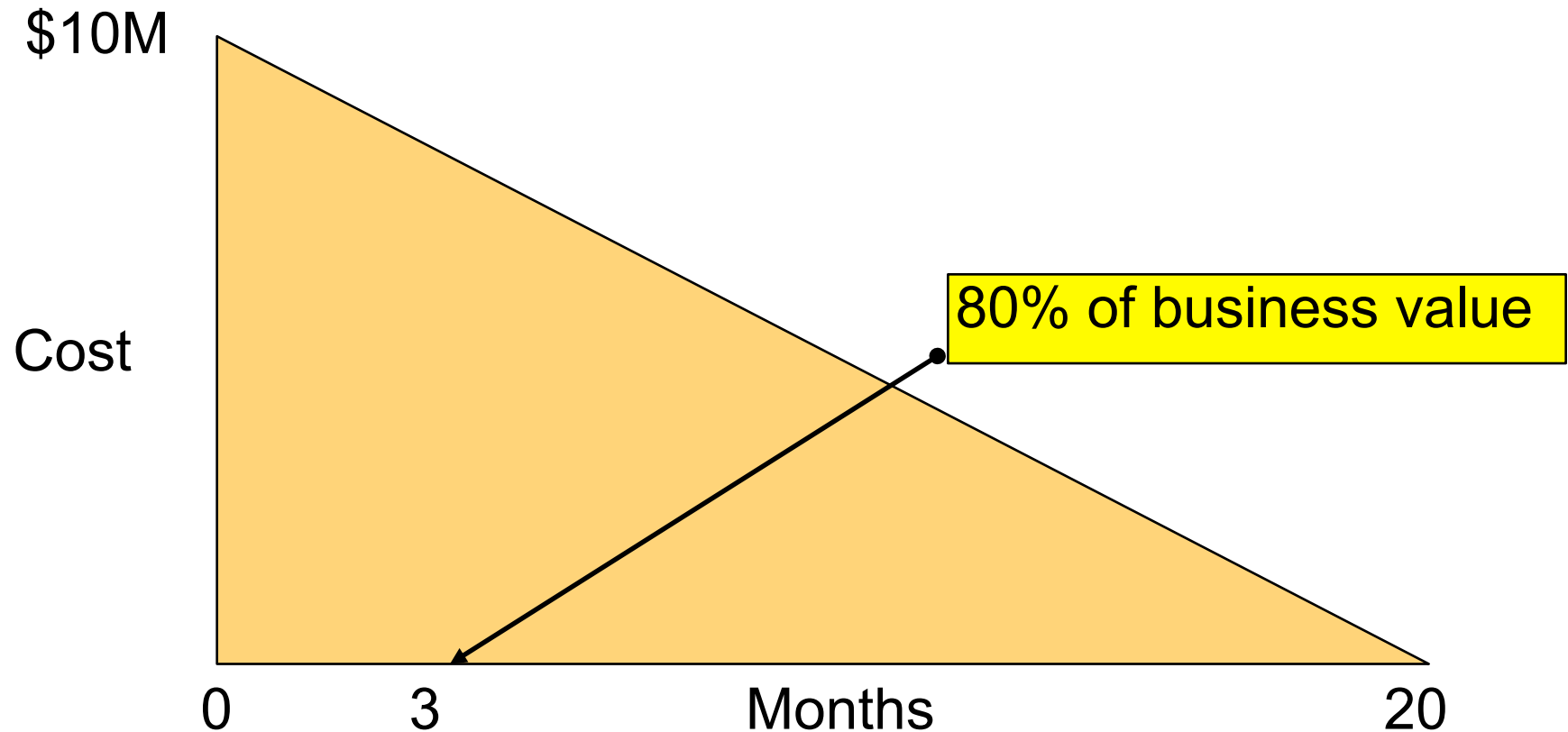
Money for Nothing and Change for Free

Development plan:

1. **Product Owner involvement allows us to tune the system to the latest known business value.**
2. **Any requirement that hasn't already been worked on can be swapped out for another of equal value;**
3. **Priority of requirements can be changed by Product Owner;**
4. **Product Owner may request additional releases at any time at prevailing time and material schedules;**
5. **Product Owner terminates development and releases product as soon as value of next feature**

© Jeff Sutherland 1993-2010

Project Management Software for Construction Company - \$10M



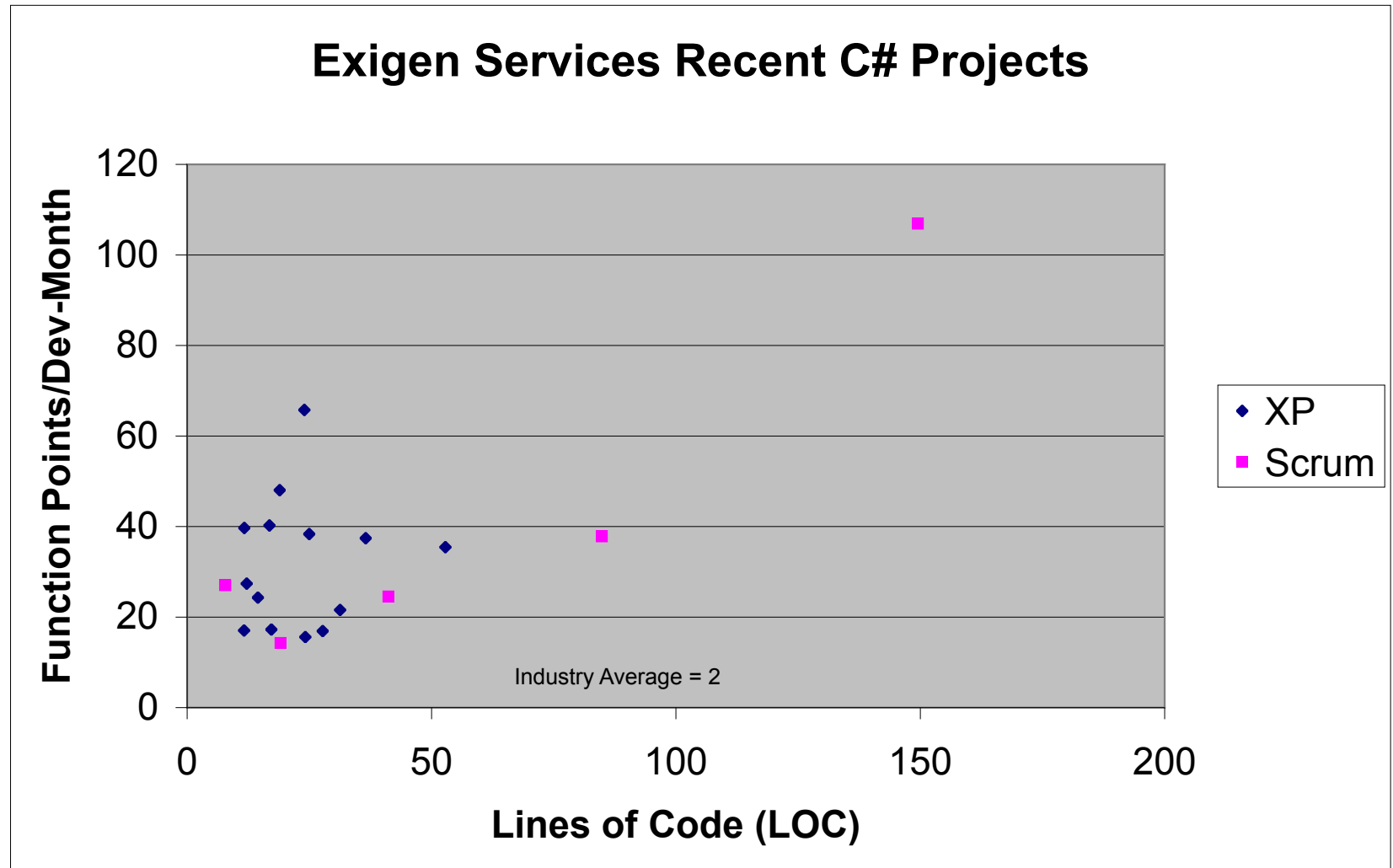
© Jeff Sutherland 1993-2010

Early Termination

Money for Nothing!

- **15% of \$10M = \$1.5M**
- **20% of \$8.5M = \$1.7M**
- **Total = \$3.2M**
- **Cost to build = \$1.3M**
- **Margin 15% → 60%**
- **Earnings increase by 400%**
- **Early retirement strategy**

Russian projects velocity data suggests high velocity is not an accident



© Jeff Sutherland 1993-2010

Exigen Services

- **Over 2000 developers**
- **Agile division in St. Petersburg has virtually all hyperproductive teams**
- **“Money for Nothing” is strategic imperative to capture value of high velocity production**
- **Requires major training and upgrade of procedures for engaging with customers in management, marketing, and deployment groups.**
- ***Disruptive technology* for dismantling worldwide waterfall market of late projects over budget with unnecessary features, poor**

Results of Customer Research

- **Set up CIO dinners in London and New York for 50 people.**
- **1/3 of attendees ready to start contracting using “Money for Nothing” strategy.**
- **1/3 ready to start talking about using this strategy and wanted sales team engagement.**
- **1/3 said their organizations were too dysfunctional to execute this strategy**
 - could not get good product backlog
 - could not prioritize features by value
 - lack of trust between management, development,

CIO Requests

- **CIOs want to know velocity of vendor teams before committing to long term project**
- **Early short term engagement to develop product backlog and validate development team velocity was viewed as desirable.**
- **Long term contract negotiations based on real velocity of early teams.**

Example: Flex-Agility 2.0

- Flex-Agility 2.0 is a premium product
 - It is not the lowest cost way or even the quickest way to conduct a project with Exigen
 - It does not fit all contracts
- It is a way to guarantee a delivery and still have the option for a high degree of change
- It is not for all customers
 - We may say “no” and work on T & M
- T & M is low risk so standard T & M is OK
- Flex-Agility 2.0 is more shared risk and so commands a premium
- We deliver highest business value first and so early termination with value is a real and desired outcome

Flex-Agility 2.0 Value Proposition

- **Larger projects (>500K) where customer wants guarantees or shared risk with Agile flexibility**
- **Buy vs. Build**
 - Certainty of Buy but with bespoke Build
- **Guaranteed velocity and estimates**
 - Commercial penalties for underachieving velocity
- **Best endeavors to correct problems**
- **Business value rather than head count tracking and billing**
- **Option of closing early should enough business value be achieved – this is “Money for Nothing”**
- **Option of adding new requirements into scope during project by replacing with lower priority requirements of equivalent “size” – this is “Change for Free”**

Exigen Services White Paper Outlining Next-Generation Outsourcing Engagement Model

San Francisco, July 8, 2008 – Exigen® Services, the leading next-generation application outsourcing provider, today announced the availability of a free white paper that details how fixed price Agile in a distributed outsourcing environment works. The white paper, titled “Unlimited Change for A Fixed Price: the Next Generation of Outsourcing Contracts” provides the framework for establishing a truly collaborative model that further aligns IT and the business with their outsourcing provider.

© Jeff Sutherland 1993-2010

Recommendations

■ ScrumButt

- Stick to time and materials body-shopping with low margins
- Work hard for the rest of your life



■ Hyperperforming teams

- Monetize your performance
- For five times the velocity, get five times the margins
- Use “Money for Nothing and Change for Free” strategy



■ Make the world a better place by altering the fundamental structure of the IT industry

- Implement the design goal of Scrum, bring all projects in early, disrupt waterfall competitors, and execute the early retirement plan!

© Jeff Sutherland 1993-2010

Questions?



© Jeff Sutherland 1993-2010