

A group of athletes in white and orange uniforms running a race. The image is slightly blurred, suggesting motion. The athletes are in various stages of a running stride, with some in the foreground and others in the background. The background is dark and out of focus.

The Roots of Scrum:

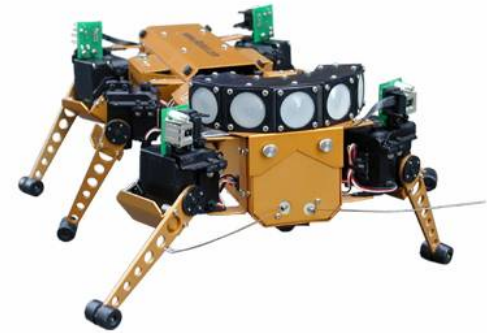
How the Japanese experience changed global software development

With help from Microsoft, Yahoo, Adobe, GE, Oracle, Siemens, BellSouth, Unicom, Palm, St. Jude Medical, DigiChart, Healthwise, Sony/Ericson, Accenture, EOS, Systematic, Starsoft Labs, SirsiDynix, Softhouse, Philips, Barclays Global Investors, Constant Contact, Wellogic, Medco

Jeff Sutherland, Ph.D.
Co-Creator of Scrum
<http://jeffsutherland.com/scrum>

Roots of Scrum

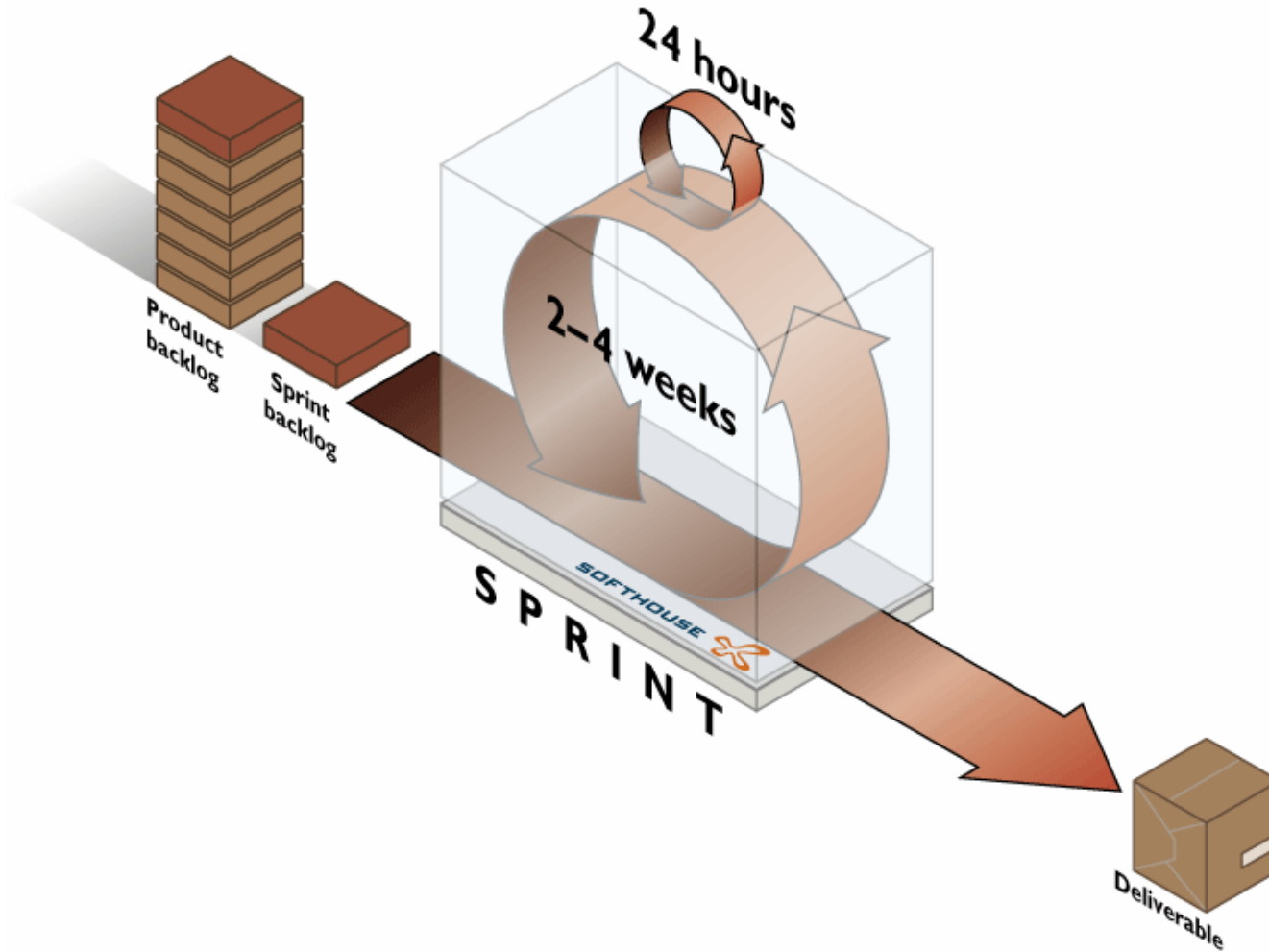
- Making the world a better place
 - Japanese manufacturing - Edward Deming
 - Team process – Silicon Valley entrepreneurs
 - Micro enterprise development – Accion and Grameen Bank
- Process and productivity research
 - Complex adaptive systems
 - IBM Surgical Team (Mythical Man Month)
 - iRobot – subsumption architecture



Scrum – *value driven* not plan driven

- Empower lean teams to deliver more software earlier with higher quality.
- Demonstrate working features to the customer early and often so the customer can inspect progress and prioritize change.
- Deliver exactly what the client wants by directly involving the customer in the development process.
- Provide maximum business value to the customer by responding to changing priorities in real time.

Scrum – inspect and adapt framework

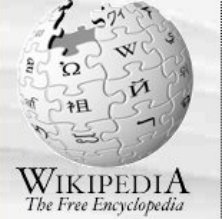


A company is a complex adaptive system so Scrum requires complex adaptive behavior

- Self organization
- No single point of control
- Interdisciplinary teams
- Emergent behavior
- Outcomes emerge with high dependence on relationship and context
- 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.



Subsumption Architecture

- **Subsumption architecture** is an AI concept originating from behavior based robotics. This term was invented by Rodney Brooks and colleagues in the mid to late 1980s.
- A subsumption architecture is a way of decomposing complicated intelligent behaviour into many "simple" behaviour modules, which are in turn organized into layers. Each layer implements a particular goal of the agent, and higher layers are increasingly more abstract. Each layer's goal subsumes that of the underlying layers, e.g. the decision to move forward by the eat-food layer takes into account the decision of the lowest obstacle-avoidance layer.
- For example, a robot's lowest layer could be "avoid an object", on top of it would be the layer "wander around", which in turn lies under "explore the world". The top layer in such a case could be "create a map", which is the ultimate goal. Each of these horizontal layers accesses all of the sensor data and generates actions for the actuators — the main caveat is that separate tasks can suppress (or overrule) inputs or inhibit outputs. This way, the lowest layers can work like fast-adapting mechanisms (reflexes), while the higher layers control the main direction to be taken in order to achieve the overall goal. Feedback is given mainly through the environment.

- R. A. Brooks (1986) "A Robust Layered Control System For A Mobile Robot", IEEE Journal Of Robotics And Automation, RA-2, April. pp. 14-23.
- R. A. Brooks (1987). "Planning is just a way of avoiding figuring out what to do next", Technical report, MIT Artificial Intelligence Laboratory.
- R. Brooks and A. Flynn (1989), "Fast, cheap, and out of control: A robot invasion of the solar system," J. Brit. Interplanetary Soc., vol. 42, no. 10, pp. 478-485, 1989. (The paper later gave rise to the title of the film Fast, Cheap and Out of Control, and the paper's concepts arguably have been seen in practice in the 2003/4 Mars Exploration Rover Mission.)

Scrum – Roles, Ceremonies, Artifacts

- Teams (7 ± 2) working together in Sprints to produce working code at Sprint Review
- 3 Roles
 - Product Owner
 - Team
 - ScrumMaster
- 3 Ceremonies
 - Sprint Planning
 - Daily Scrum
 - Sprint Review
- 3 Artifacts
 - Product Backlog
 - Sprint Backlog
 - Sprint Burndown

Requires team
based incentives



Dr. W. Edward Deming

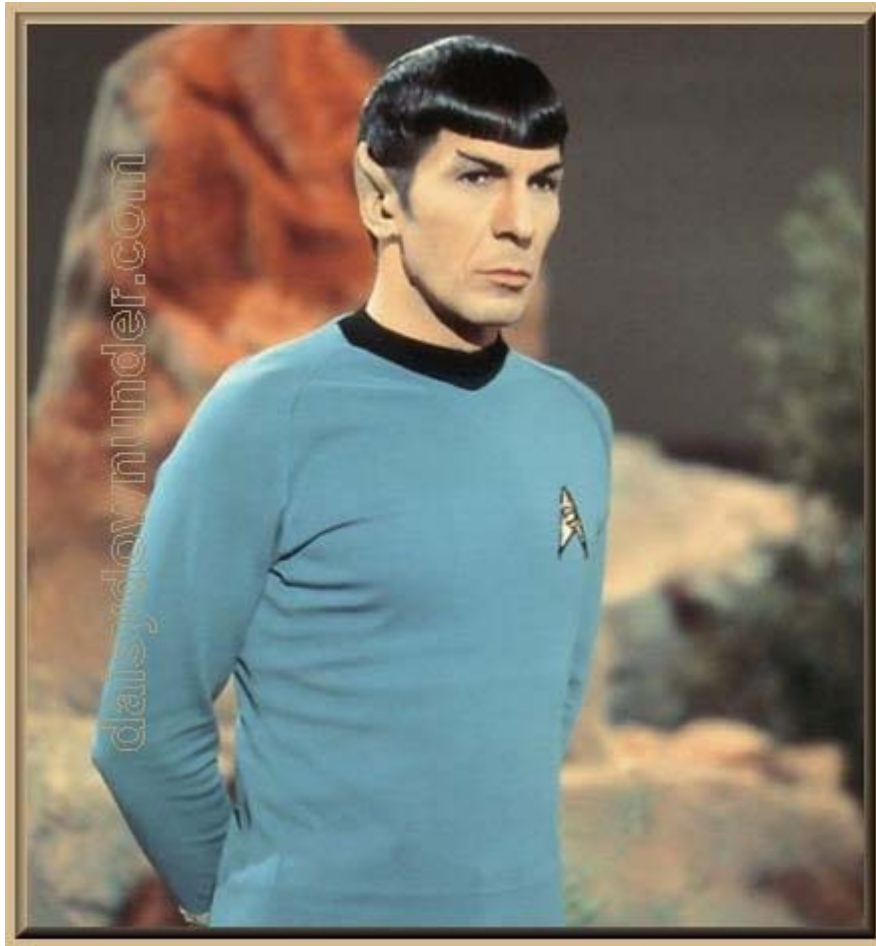
- Father of Japanese post-war industrial revival and leading quality guru in both Japan and the United States. In the 1940's he vigorously promoted iterative development.

- “Our prevailing system of management has destroyed our people. People are born with intrinsic motivation, self-respect, dignity, curiosity to learn, joy in learning. The forces of destruction begin with toddlers—a prize for the best halloween costume, grades in school, gold stars—and on up through the university. On the job, people, teams, and divisions are ranked, rewarded for the top, punished for the bottom. Management by Objectives, quotas, incentive pay, business plans, put together separately, division by division, cause further loss, unknown and unknowable.”
Edward Deming in letter to MIT Prof. Peter Senge

- “I believe that the prevailing system of management is, at its core, dedicated to mediocrity. It forces people to work harder and harder to compensate for failing to tap the spirit and collective intelligence that characterizes working together at its best.”
Prof. Peter Senge response

Larman, Craig and Basili, Vic. Iterative and Incremental Development: A Brief History. IEEE Computer, [June 2003 \(Vol. 36, No. 6\)](#) pp. 47-56
Senge, P. M. (2006). *The Fifth Discipline: The Art and Practice of the Learning Organization*, Revised Edition. New York, Doubleday.

It's illogical!



- I don't understand these Americans but they are very interesting.
- As for Toyota, resistance is futile.

Toyota Motor Manufacturing North America Mission

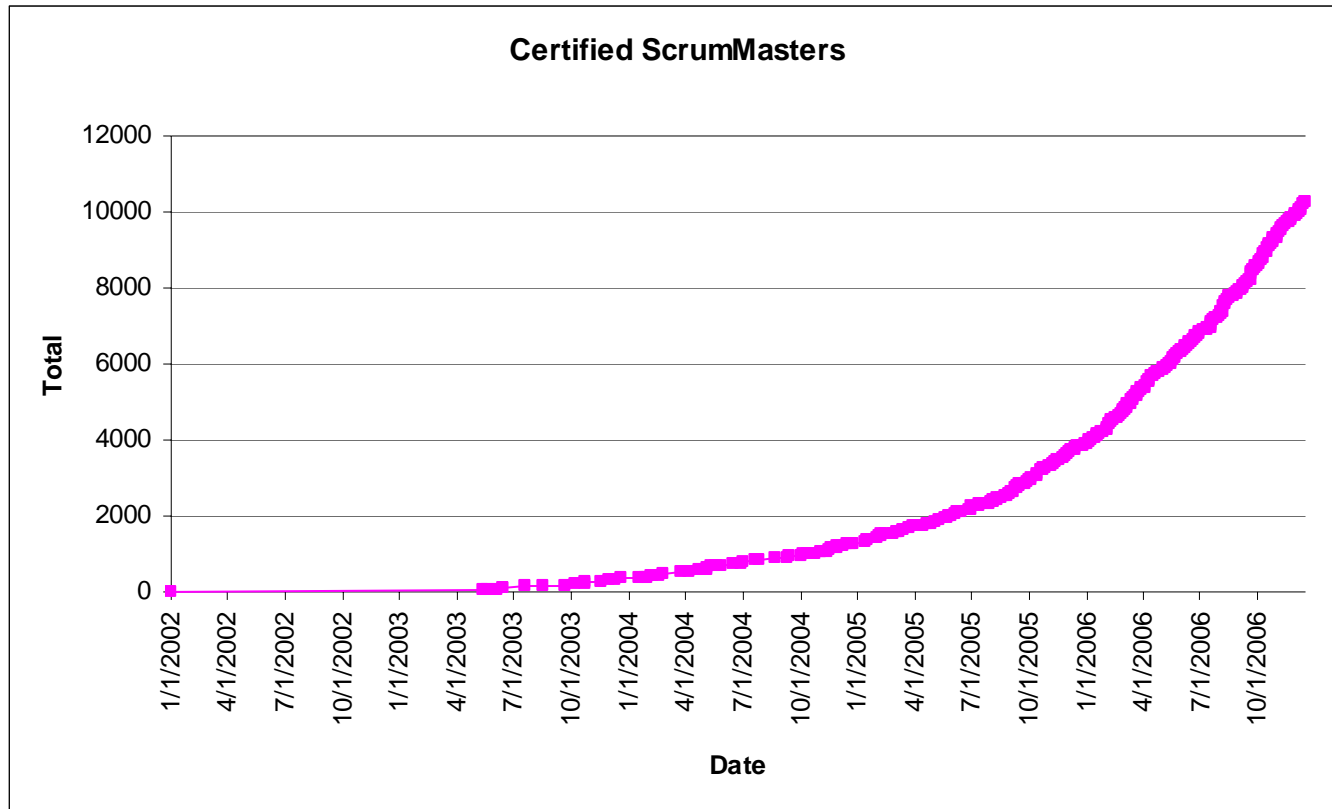
1. As an American company, contribute to the economic growth of the **community** and the United States.
2. As an independent company, contribute to the **stability and well-being of team members**.
3. As a Toyota group company, contribute to the overall growth of Toyota by **adding value to our customers**.

Scrum Mission Statement

- Build communities of stakeholders (customers, companies, development teams) that increase the economic well-being of all concerned.
- Enhance development team work environments by empowering people to work together in more creative, innovative, and productive ways.
- Deliver the highest possible customer value in the shortest possible time to improve the customers work experience. Make systems easier to use, more helpful to the user, and more fun to experience.

Yahoo Chief Product Owner – “Scrum is faster, better, cooler! It’s the way we first built software at Yahoo, yet is scalable to large, distributed, and outsourced teams.”

Scrum Community of Practice



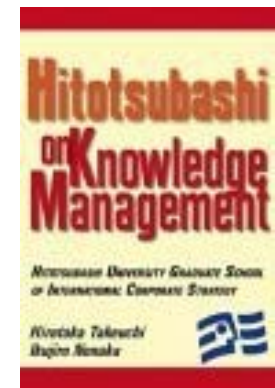
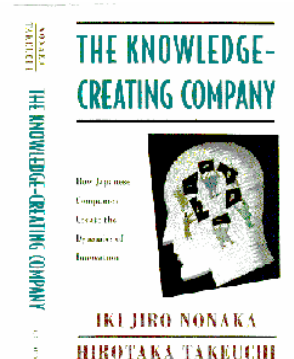
Computerworld estimates that over 2/3 of Internet projects in the U.S. use Agile methods, about 167,000 projects. (Sliwa, 2002)

Godfathers of Scrum: Hirotaka Takeuchi and Ikujiro Nonaka

Takeuchi, Hirotaka and Nonaka, Ikujiro. 1986. The new new product development game. Harvard Business Review 64:1:137-146 (Jan/Feb), reprint no. 86116.

Takeuchi, Hirotaka and Nonaka, Ikujiro. 1995. The Knowledge-Creating Company : How Japanese companies create the dynamics of innovation. New York, Oxford University Press.

Takeuchi, Hirotaka and Nonaka, Ikujiro. 2004. Hitotsubashi on Knowledge Management. Singapore: John Wiley & Sons (Asia).



Theory: Scrum Origins

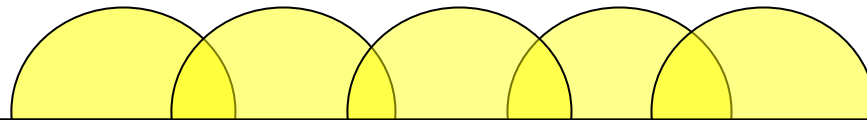
Project Management Styles

Requirements Analysis Design Implementation Testing



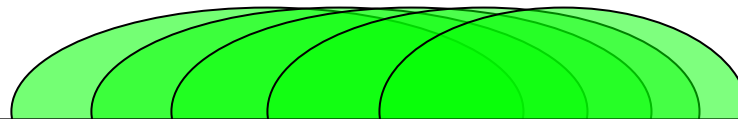
Type A – Isolated cycles of work

NASA Waterfall



Type B – Overlapping work

Fuji-Xerox Scrum



Type C – All at once

Honda Scrum

The overlapping of phases does away with traditional notions about division of labor. Takeuchi and Nonaka (1986)

Rugby Scrum

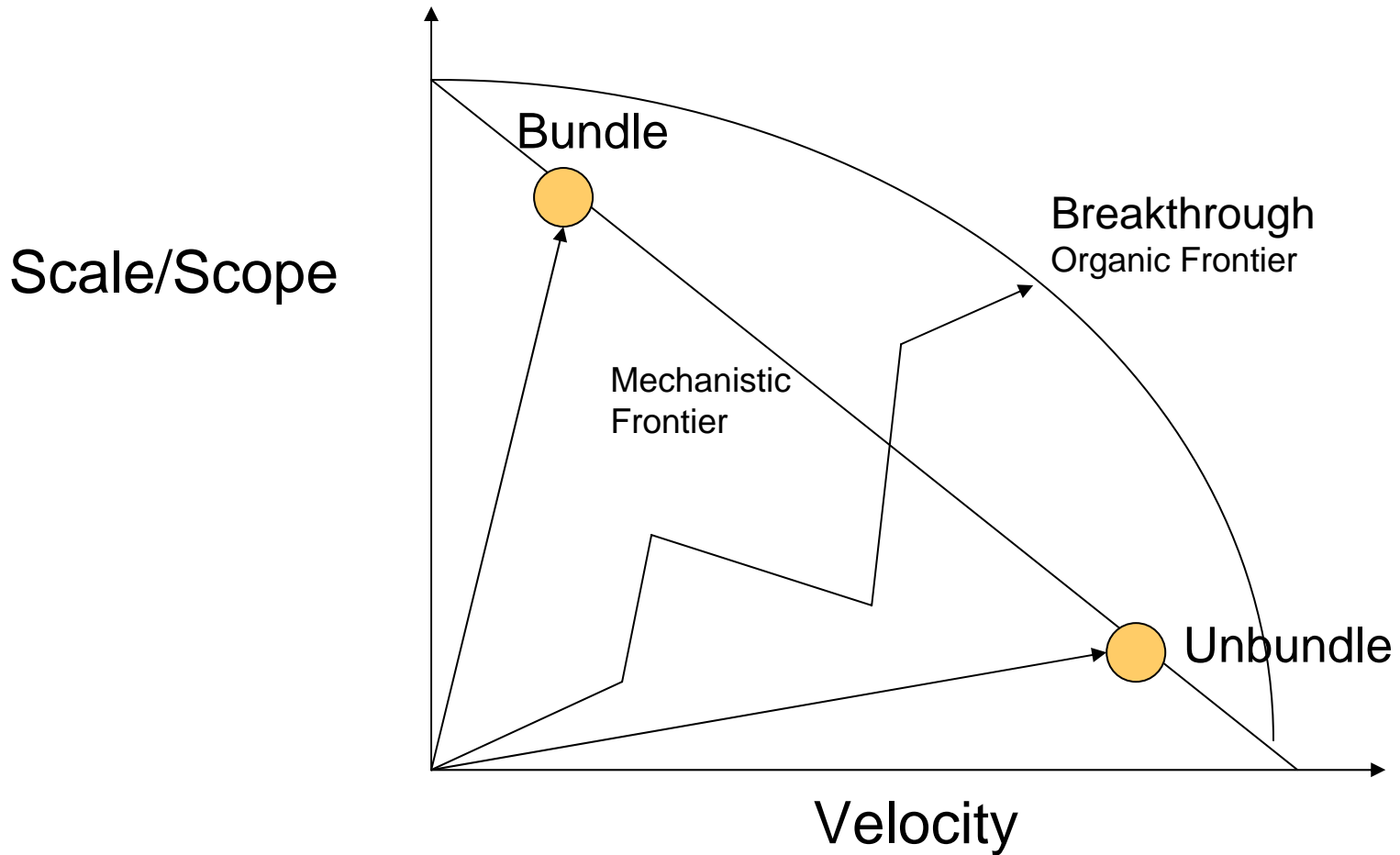


Toyota synthesis of constraints

- Historical assumption is that high quality, product variety, and low cost cannot be achieved simultaneously.
- Toyota production system is based on totally different way of thinking.
- Through knowledge creation by synthesis of contradictions, Toyota pushes the envelope.
- High quality, high variety, and low cost all at once.

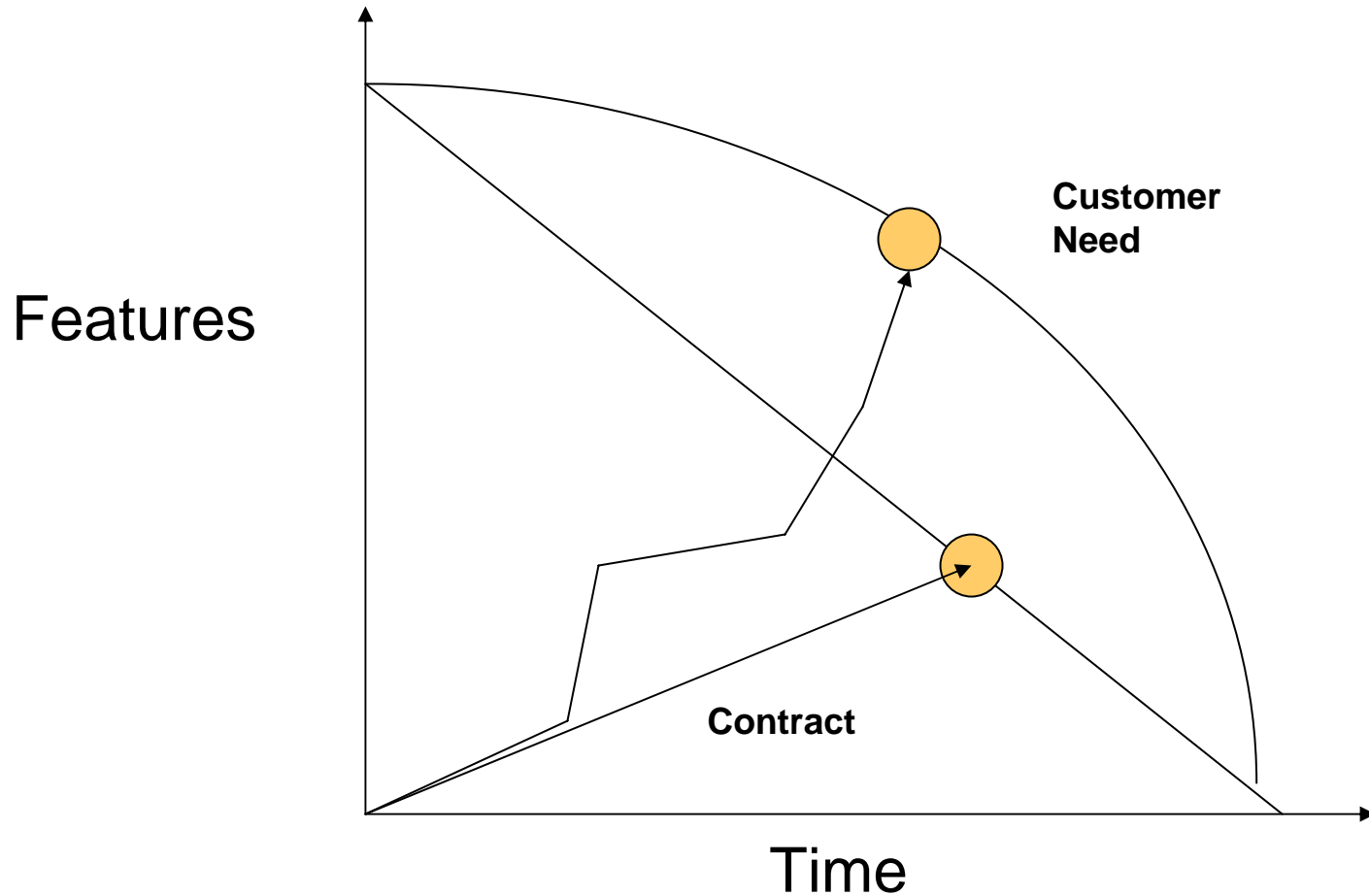
Synthesize, not Optimize

(Demings message to the Japanese)



Takeuchi, Hirotaka and Nonaka, Ikujiro. 2004. Hitotsubashi on Knowledge Management. Singapore: John Wiley & Sons (Asia).

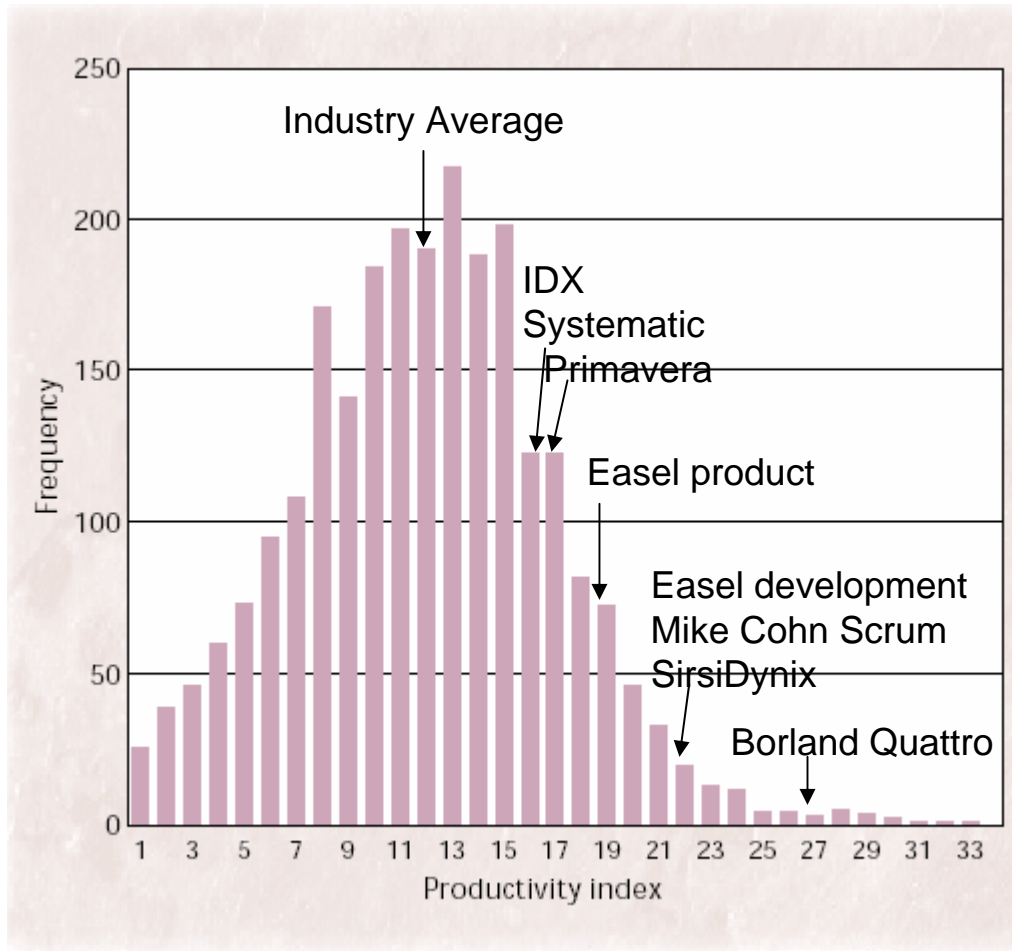
Prioritize and reprioritize Product Backlog to deliver business value



Scrum cuts through cost, time, functionality barriers

- Agile process – adapt and inspect
- Iterative, incremental – close to customer
- Used to manage complex projects since 1990;
- Delivers business functionality every 30 days;
- Extremely simple but very hard

We simultaneously want FASTER: Putnam Process Productivity Index



■ $N = 1.27^{*(N-1)}$

■ 33 is 2098 times 1

Myers, W., Why Software Developers Refuse to Improve. Computer, 1998. 31(4) 110-112

And BETTER

- Productivity – product backlog requirements completed per 100,000 investment

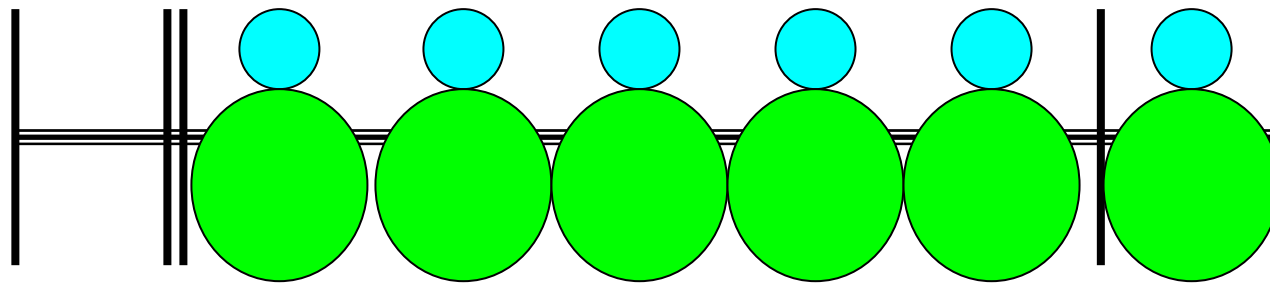
Months since Type B Scrum implemented	3	12	24
Productivity	4.5	9.0	12.2
Quality	100+	100	5

Source: Primavera

And MORE for LESS

1. Welcome changing requirements, even late in development.
2. Deliver working software frequently.
3. Business people and developers work together daily.

Scrum is ITERATIVE, customer can CHANGE requirements, and solution EMERGES through self-organization



Scaling

Planning

Developing

Deploying

Cultural Change - Japanese style

<i>Old Organization</i>	<i>New Organization</i>
Centralized	Distributed
Unified perspective	Diversified perspective
Original meaning	Emergent meaning
Analytical	Creative
Analysis to action	Learning by doing
Rational	Redundant
Certain	Uncertain
Strategy concept	Local action
Authoritative	Participative
Hierarchical	Flat

Breaking down command and control

- Intended strategy is developed centrally. Emergent strategy self-organizes through local actions.
 - Distributed cognition and actions
- Scrum team must be allowed to self-organize
 - Autonomous
 - Transcendent
 - Cross-fertilization
- Team chooses own work
 - Individuals manage their own work
 - Management gets out of the way

Distributed Organization - Google Way

- When Rosing started at Google in 2001, "we had management in engineering. And the structure was tending to tell people, No, you can't do that." So Google got rid of the managers. Now most engineers work in teams of three, with project leadership rotating among team members. If something isn't right, even if it's in a product that has already gone public, teams fix it without asking anyone. *Agile Principle #5, 9, 12*
- "For a while," Rosing says, "I had 160 direct reports. No managers. It worked because the teams knew what they had to do. That set a cultural bit in people's heads: You are the boss. Don't wait to take the hill. Don't wait to be managed." *Agile Principle #1, 3*
- And if you fail, fine. On to the next idea. "There's faith here in the ability of smart, well-motivated people to do the right thing," Rosing says. "Anything that gets in the way of that is evil." *Agile Principle #5, 12*

Fast Company. April, 2003.

Diversified perspective

- Cross-functional teams
- Scrum team has product knowledge, business analysts, user interface design, software engineers, QA
- Advanced Scrum pulls in all stakeholders – management, customers, installation, and support.

Toyota Prius – emergent strategy

- Revolution in product, technologies, and process
 - Does not fit any product line. Designed for new perspective.
- Uses many technologies
 - Engine, motor, battery, braking combine into hybrid system
- Developed in record time
 - 15 months instead of four years
- Overlapping phases
 - Research, development, design, production
- Leaders built, utilized, and energized “*ba*”

The concept of *ba*

- Dynamic interaction of individuals and organization creates a synthesis in the form of a self-organizing team.
- It provides a shared context in which individuals can interact with each other.
- Team members create new points of view and resolve contradictions through dialogue.
- *Ba* is shared context in motion where knowledge as a stream of meaning emerges.
- *Emergent knowledge codified into working software self-organizes into a product.*

Prius project team managed “*Ba*”

- Leaders can “find” and utilize spontaneously formed *ba*
- Leaders can build *ba* by providing space for interactions
 - Physical space such as meeting rooms
 - Cyberspace such as computer network
 - Mental space such as common goals
- Fostering love, care, trust, and commitment forms the foundation of knowledge creation (self-organization)
- *Scrum is based on TRUTH, TRANSPARENCY, and COMMITMENT*

Energy of ba is given by its self-organizing nature

- *Ba* needs to be “energized” with its own intention, direction, interest, or mission to be directed effectively.
- Leaders provide autonomy, creative chaos, redundancy, requisite variety, love, care, trust and commitment.
- Prius creative chaos was generated by demanding goals. Uchiyamada demanded that his team question every norm on new car development.
- Top management put Prius project team under great time pressure which caused extreme use of simultaneous engineering
- Equal access to information at all levels was critical
- *ScrumMaster and management must “energize” ba through facilitating colocation, dynamic interaction, face to face communication, transparency, and audacious goals.*

Local action forces self-organization

- Individual self-organizes work
- Team self-organizes around goals
- Architecture self-organizes around working code
- Product emerges through iterative adaptation
- Requires participative approach as opposed to authoritative approach
- Flat organizational structure

First Scrum – Easel 1993

- Abandoned GANTT charts
- Abandoned job titles
- Created ScrumMaster
- Created Product Owner
- Daily meetings to foster self-organization
- Shielded team from interference during Sprint
- Sprint planning, Sprint review, demo, retrospective
- Agnostic about engineering practices
- Used all XP engineering practices

Scrum Influence on eXtreme Programming

From: Kent Beck To: Jeff Sutherland <jsutherland>

Reply: 70761.1216@compuserve.com

Date: Mon, 15 May 1995 18:01:15 -0400 (EDT)

Subj: HBR paper

Is there a good place to get reprints of the SCRUM paper from HBR? I've written patterns for something very similar and I want to make sure I steal as many ideas as possible.

Kent

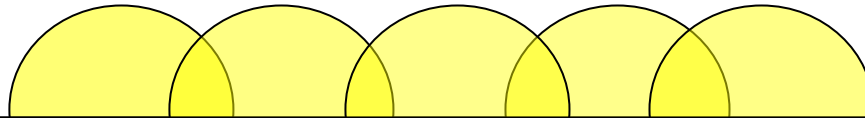
Scrum and XP

- The first hyperproductive Scrum used all the XP engineering practices.
- The highest performance large project ever documented uses Scrum and XP together.
- You cannot get a Scrum with extreme velocity without XP.
- You cannot scale XP without Scrum.
- Example: SirsiDynix project

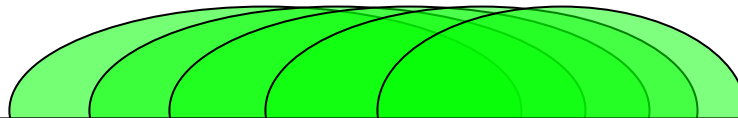
Distributed/Outsourcing Styles



Isolated Scrums



Distributed Scrum of Scrums



Totally Integrated Scrums

Outsourcing

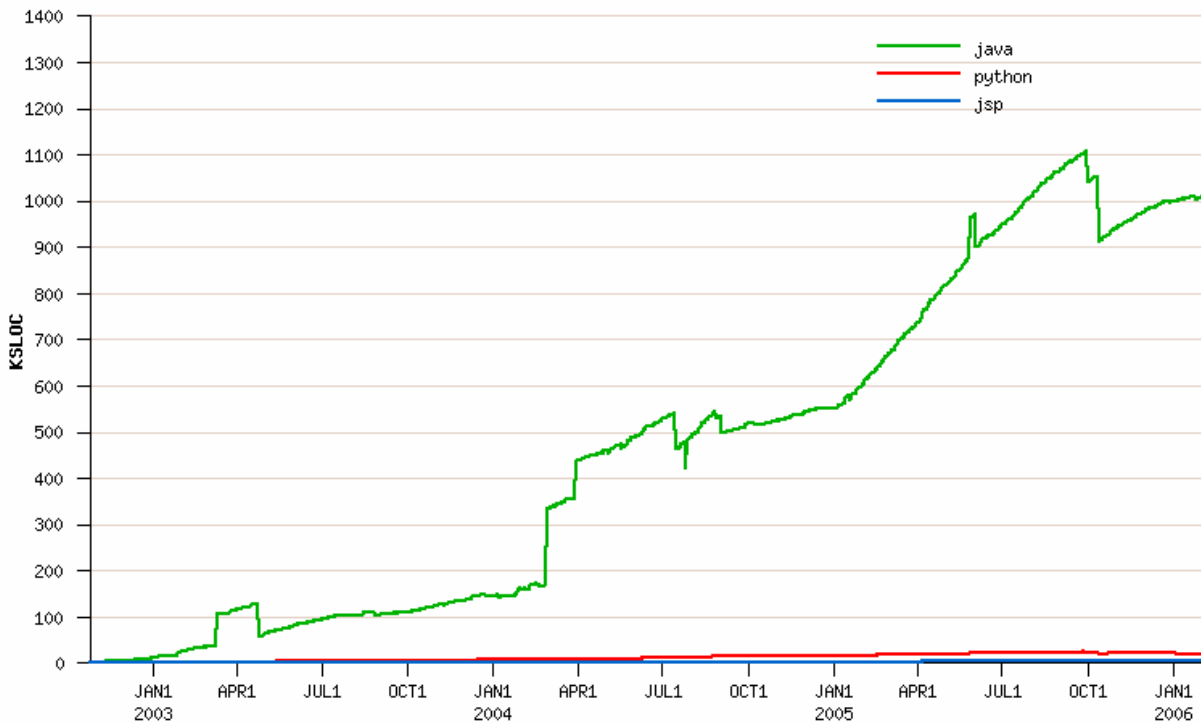
- Outsource \$2M development
- Outsourcing costs - \$1.6M
 - Industry data show 20% cost savings on average
- Introduce Scrum locally
 - 240% improvement
- Local Scrum costs – \$0.83M
- Outsourcing is appropriate only when:
 - You need expertise that is unavailable locally
 - You want to expand and contract development staff without layoffs
 - You can linearly scale productivity with Agile teams

4th Generation ILS – Version 8.0

- ILS is like a ERP system for a vertical market
- 3rd Generation ILS was a client/server app running at more than 12,000 locations in 42 countries
- More than 120 million users
- 4th Generation is a completely new development effort
 - 100% Java development
 - More than 1 million lines of source code
 - More than 3 thousand features
 - Kerberos security
 - Record ownership
 - Consortium support

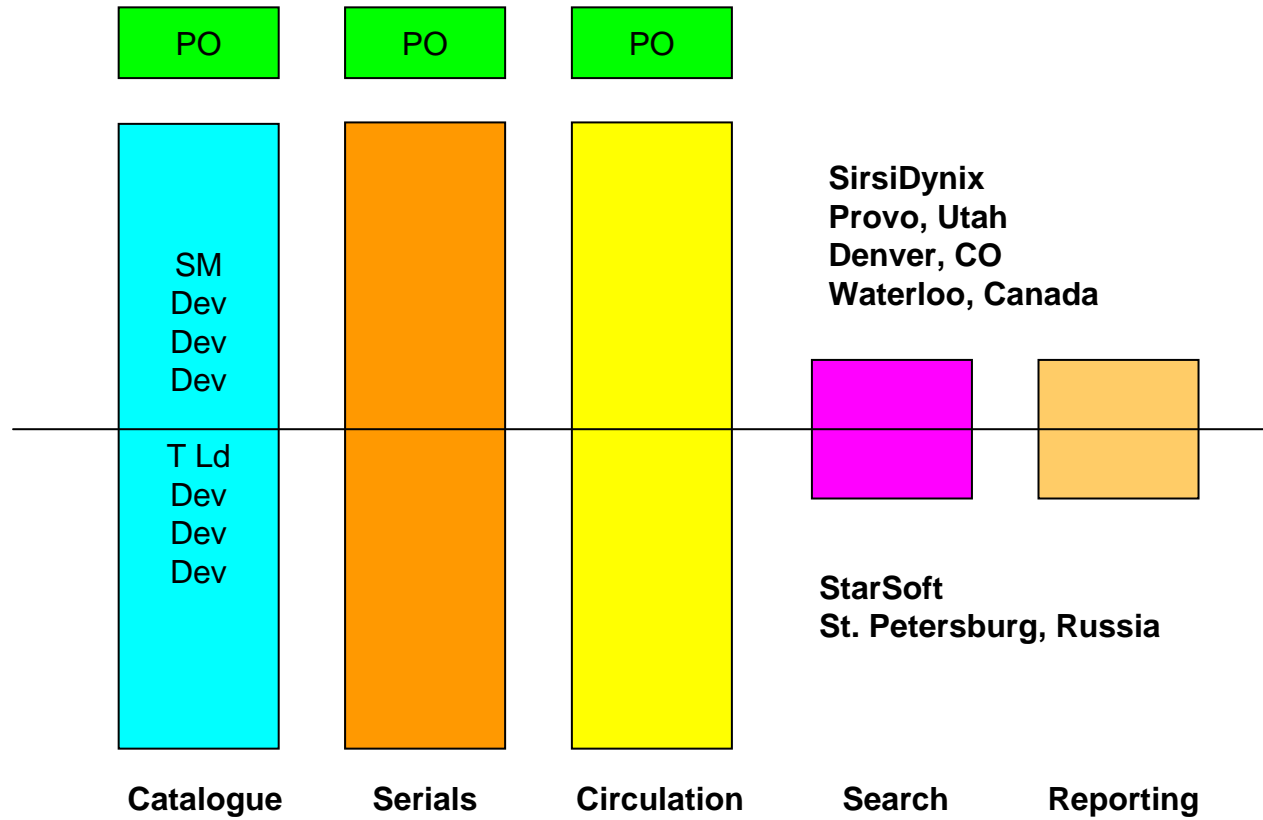
SirsiDynix Distributed Scrum

- Over a million lines of Java code



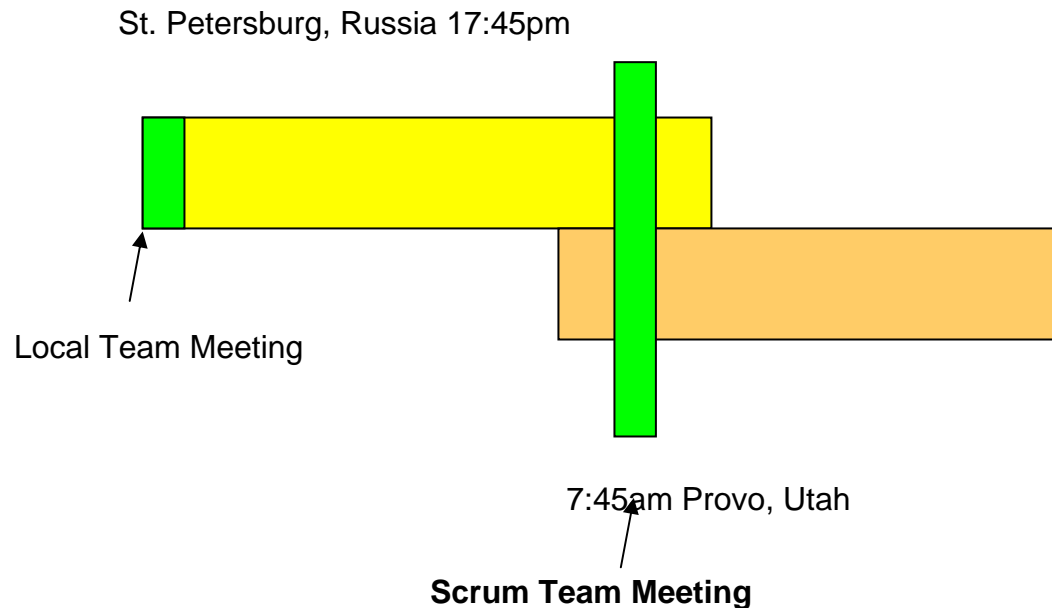
SirsiDynix Distributed Scrum

- 56 developers distributed across sites



SirsiDynix Distributed Scrum

■ Scrum daily meetings



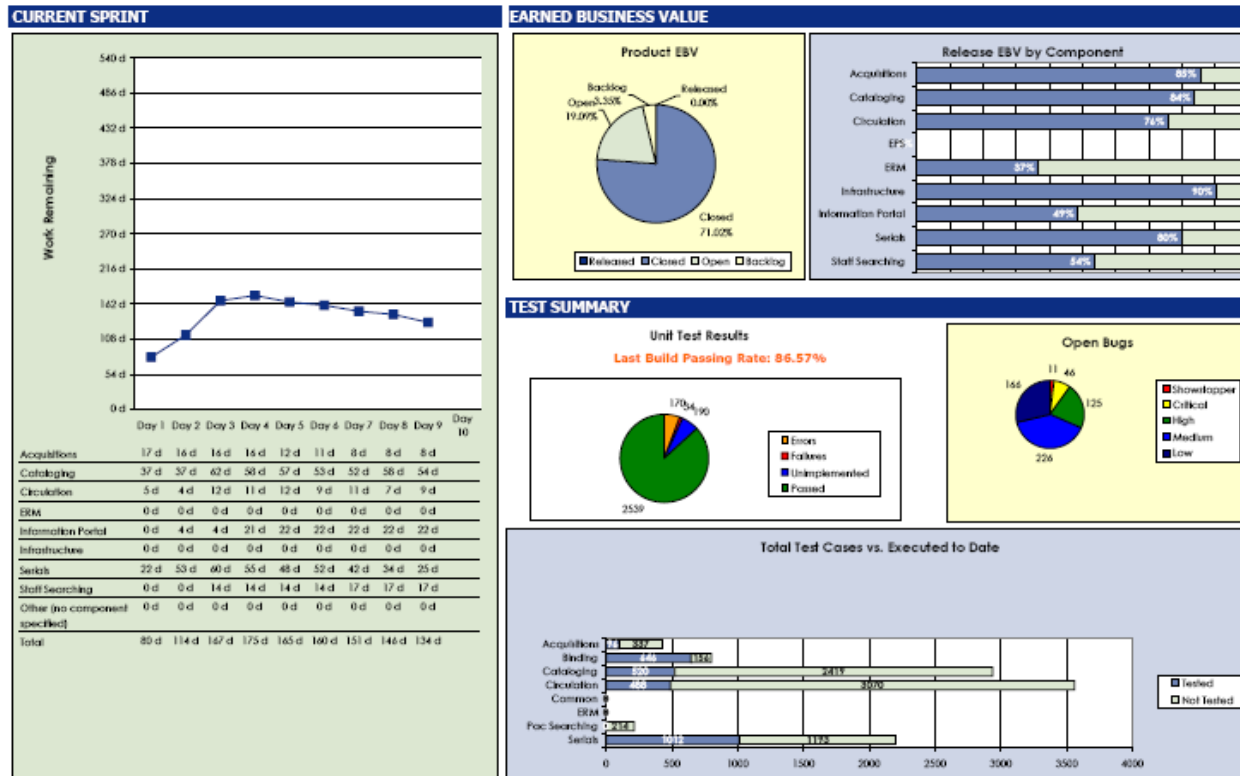
SirsiDynix Distributed Scrum

Common tools



Horizon 8.0

Report Ending: Monday, 17 Oct 2005



SirsiDyNix Distributed Scrum

- Uncommon performance

	SCRUM Small team	Waterfall	SirsiDyNix Large team
Person Months	54	540	827
Lines of Java	51,000	58000	671,688
Function Points	959	900	12673
FP per dev/month	17.8	2.0	15.3

Company Structure

Discipline Whimsical	Bureaucracy <ul style="list-style-type: none"> • Rigid rule enforcement • Extensive written rules and procedures • Hierarchy controls 	Leadership <ul style="list-style-type: none"> • Empowered employees • Rules and procedures as enabling tools • Hierarchy supports organizational learning
	Autocracy <ul style="list-style-type: none"> • Top down control • Minimum rules and procedures • Hierarchy controls 	Organic <ul style="list-style-type: none"> • Empowered employees • Minimum rules and procedures • Little hierarchy
	Coercive	Empowering

Adapted from Liker, JK (2004) The Toyota Way. McGraw Hill.



Questions?

Bibliography

- Larman, Craig and Basili, Vic. Iterative and Incremental Development: A Brief History. IEEE Computer, [June 2003 \(Vol. 36, No. 6\)](#) pp. 47-56
- Poppendieck, M. and T. Poppendieck (2006). Lean Software Development: An Implementation Guide, Addison-Wesley.
- Schwaber, K. (2004). Agile project management with Scrum. Redmond, Wash., Microsoft Press.
- Senge, P. M. (2006). The Fifth Discipline: The Art and Practice of the Learning Organization, Revised Edition. New York, Doubleday.
- Sutherland, J., A. Viktorov, et al. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams. HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii, IEEE.
- Sutherland, J. and K. Schwaber (2007). The Scrum Papers: Nuts, Bolts, and Origins of an Agile Method. Boston, Scrum, Inc.
- Takeuchi, H. and I. Nonaka (1986). "The New New Product Development Game." Harvard Business Review(January-February).
- Takeuchi, H. and I. Nonaka (2004). Hitotsubashi on Knowledge Management. Singapore, John Wiley & Sons (Asia).