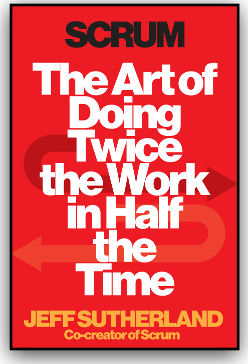
scruminc.

# Getting to Done
## The Secret Sauce of High Performing Teams

DONE

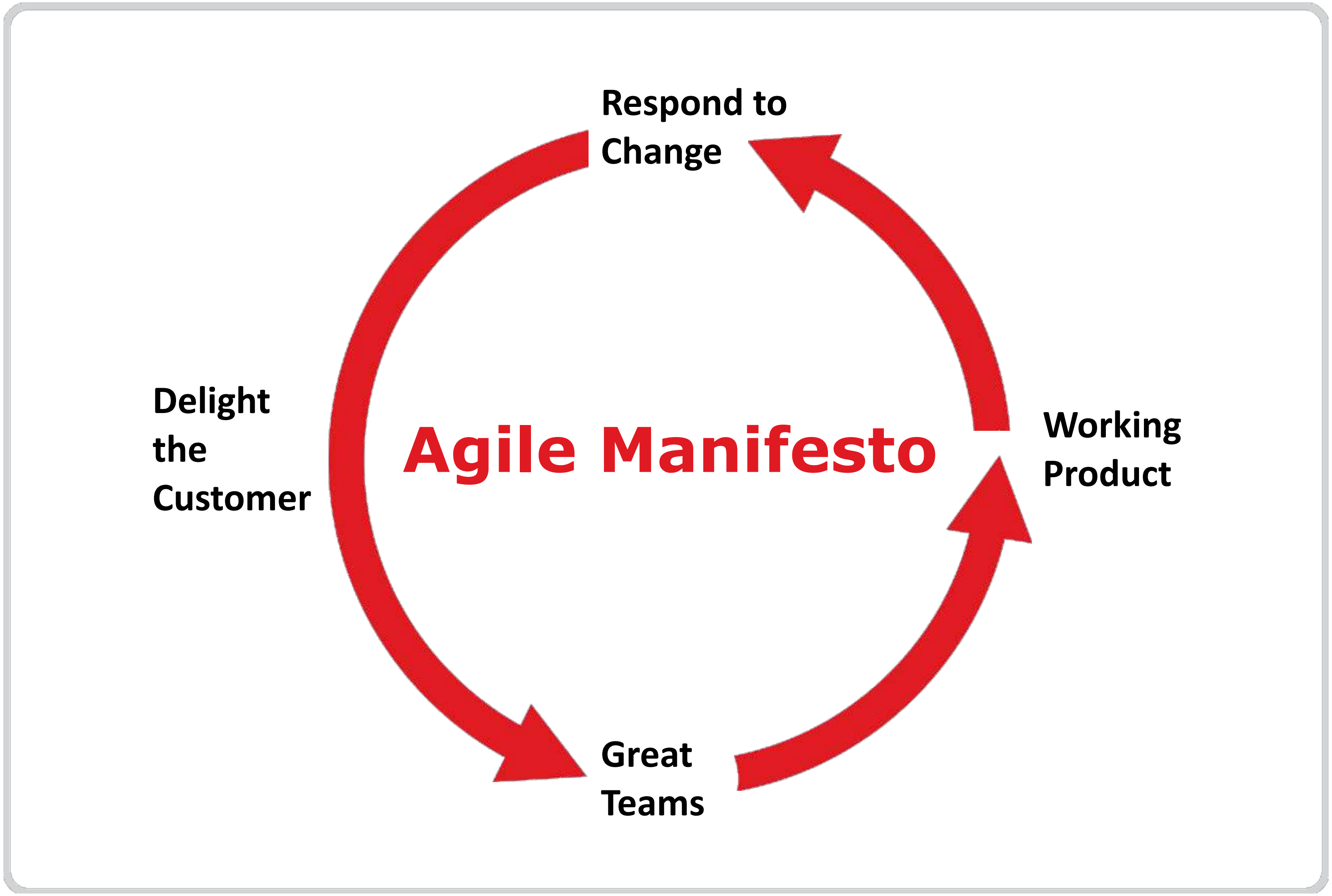
Hosts: *JJ Sutherland*
*Jeff Sutherland*

Coauthors:

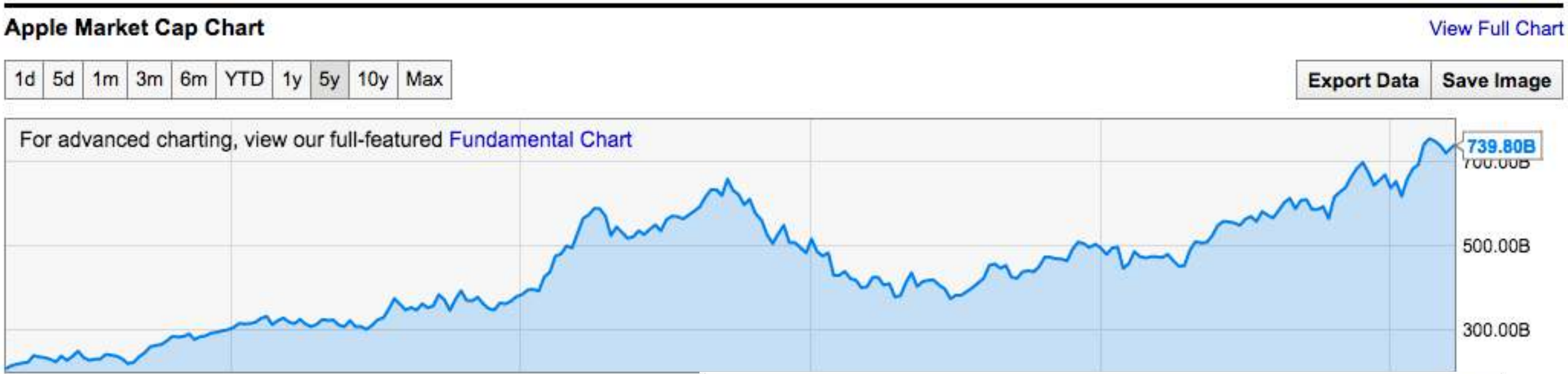


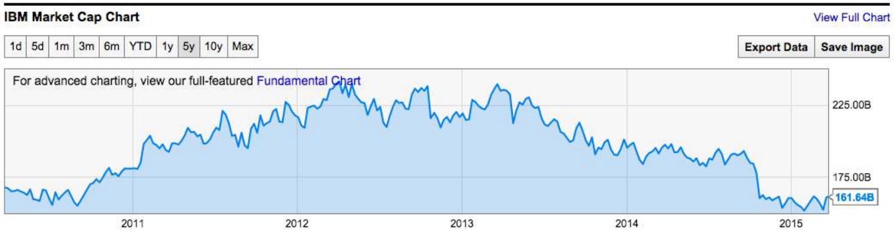
SCRUM
The Art of Doing Twice the Work in Half the Time
JEFF SUTHERLAND
Co-creator of Scrum

# Agile Means Working Product

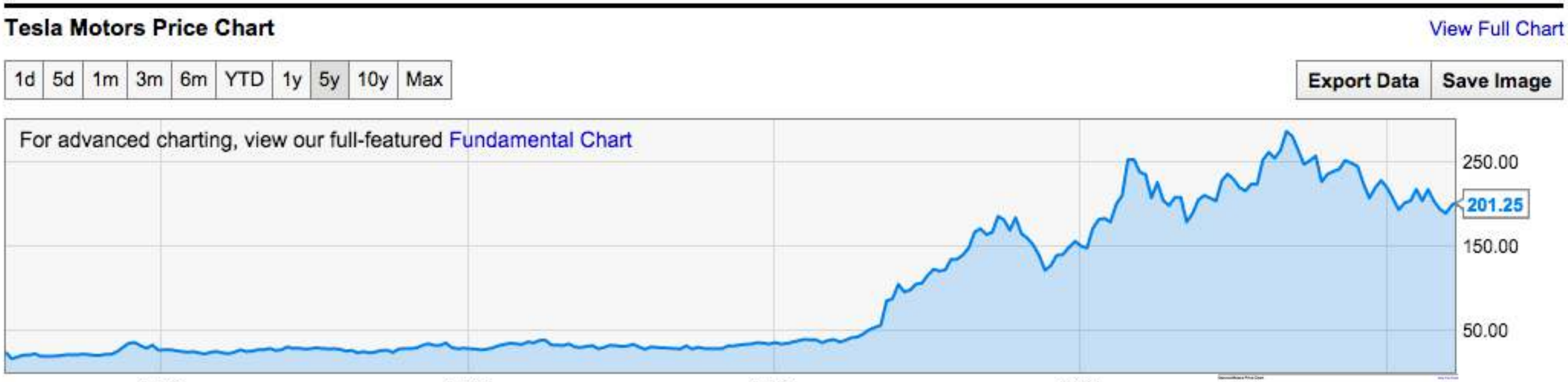


# Agile vs. Not So Agile


Apple Market Cap Chart


IBM Market Cap Chart

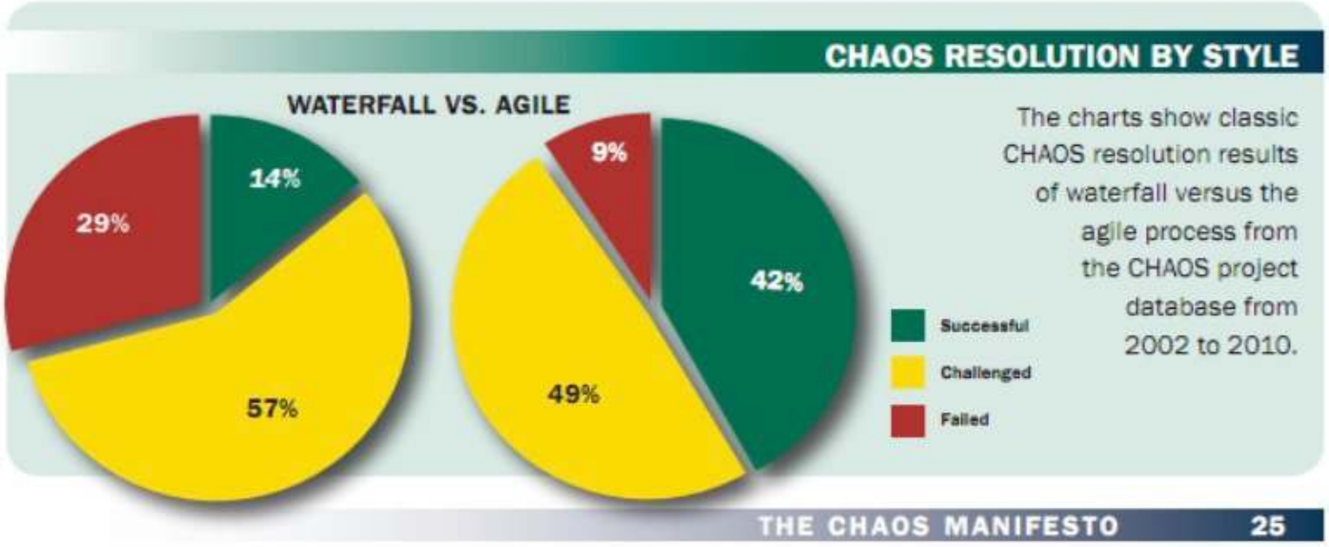
Whereas firms with a vertical mindset like IBM, are struggling with declining revenues and bloody cost-cutting reorganizations, firms in the horizontal world of Agile, like Apple and Google, are busy growing and inventing the future. *Stephen Denning 26 Jan 2015 Forbes*


Tesla Motors Price Chart

General Motors

scruminc.

# Do you have working software at the end of a sprint?



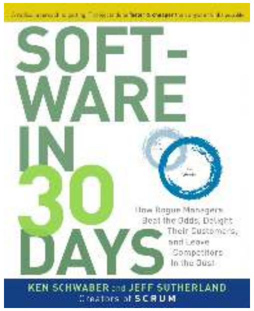
CHAOS RESOLUTION BY STYLE

The charts show classic CHAOS resolution results of waterfall versus the agile process from the CHAOS project database from 2002 to 2010.

THE CHAOS MANIFESTO 25

58% of Agile is “Bad Agile”
*More recent data as Scrum is scaling in large companies indicates this is getting worse, not better!*

Source:

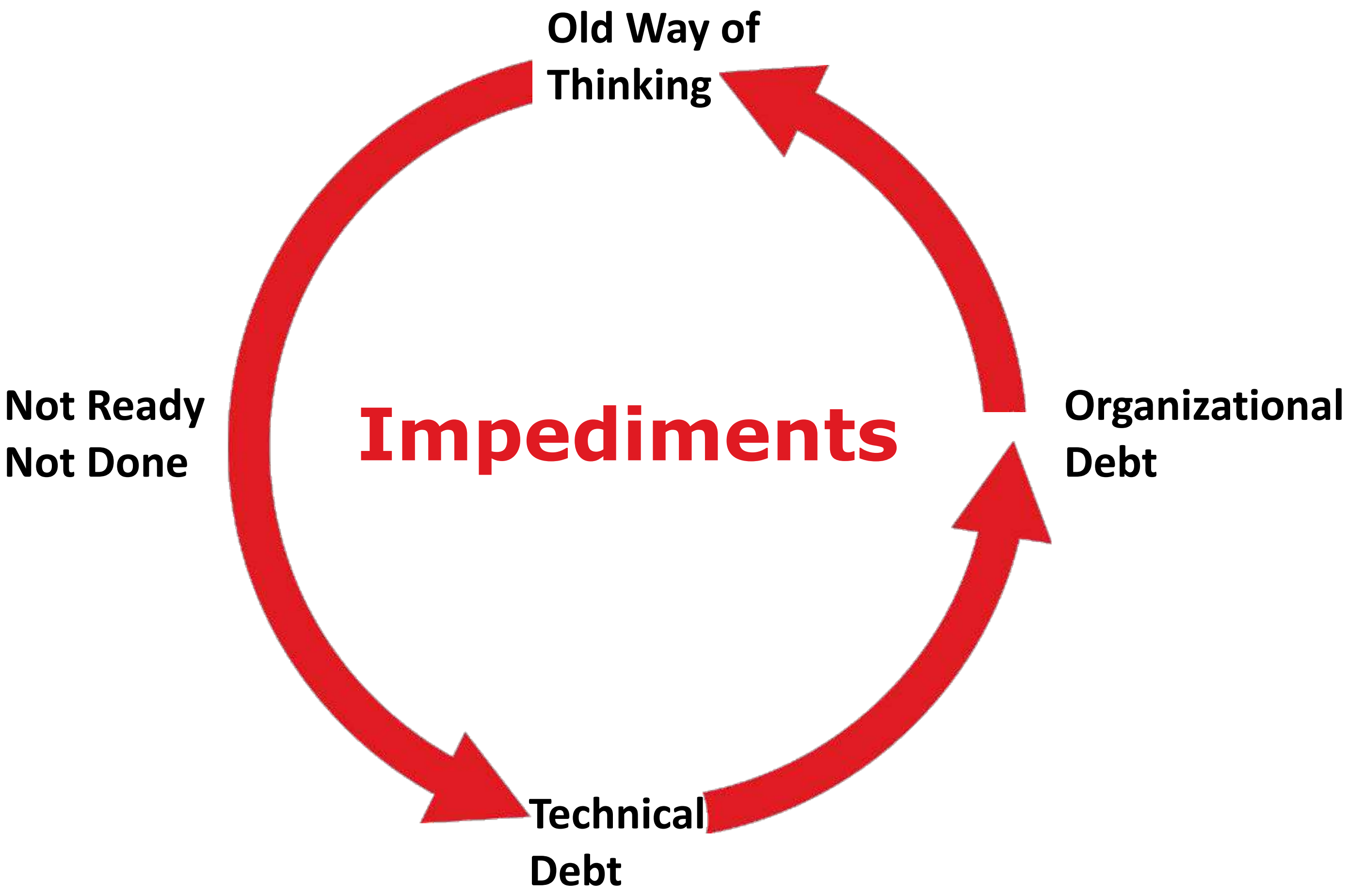


SOFT-WARE IN 30 DAYS
How Rogue Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust
KEN SCHWABER and JEFF SUTHERLAND
Creators of SCRUM

scruminc.

# Reasons for Not Done …

# The Four Horseman of the Apocalypse

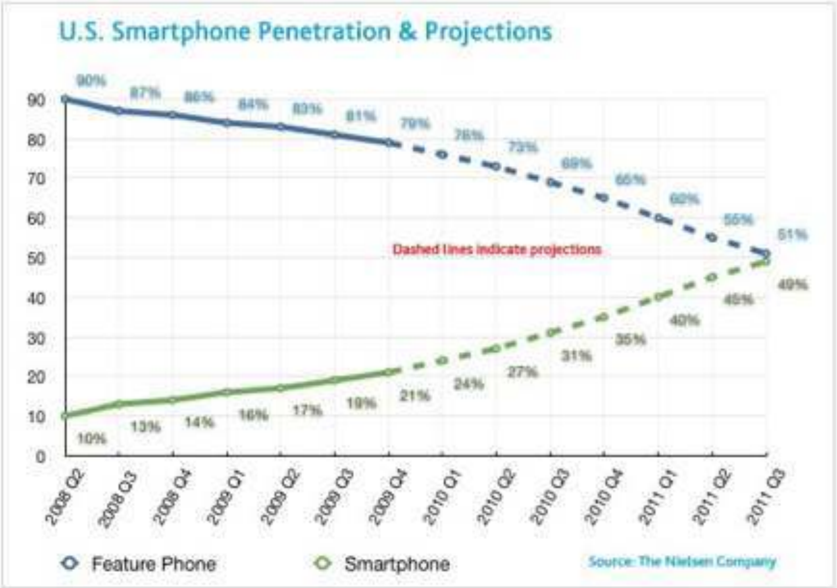


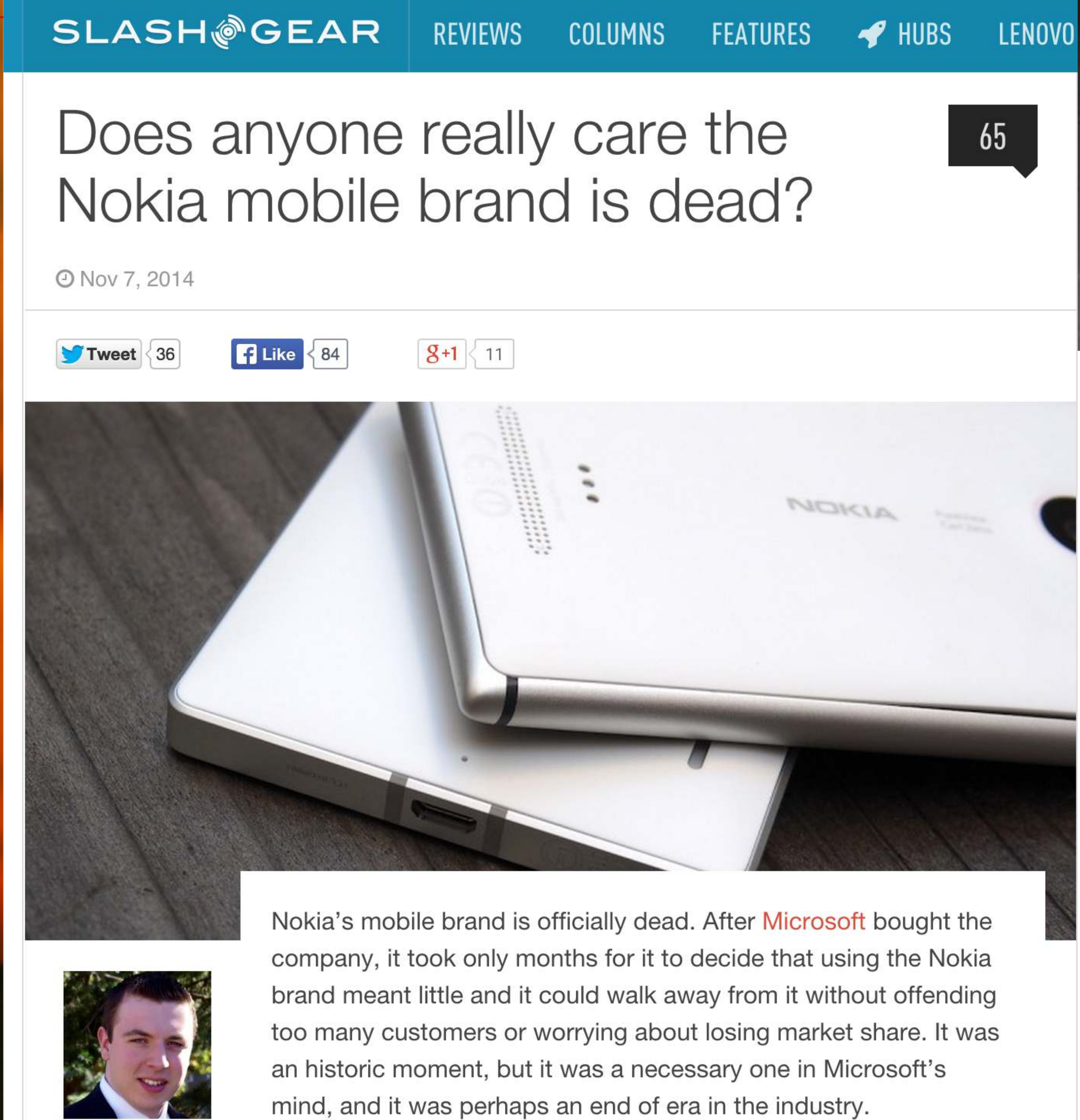
scruminc.

# Customers Often Don't Know What They Want Until They See It! *Humphrey's Law*

Customers loved this…

…Until they tried this…





# Does anyone really care the Nokia mobile brand is dead?

Nov 7, 2014

Nokia's mobile brand is officially dead. After Microsoft bought the company, it took only months for it to decide that using the Nokia brand meant little and it could walk away from it without offending too many customers or worrying about losing market share. It was an historic moment, but it was a necessary one in Microsoft's mind, and it was perhaps an end of era in the industry.

# Why Is It So Important to the Team to Have Stuff Done?

Teams That Finish Early Accelerate Faster!

# Patterns for Coaches - ScrumPlop.org

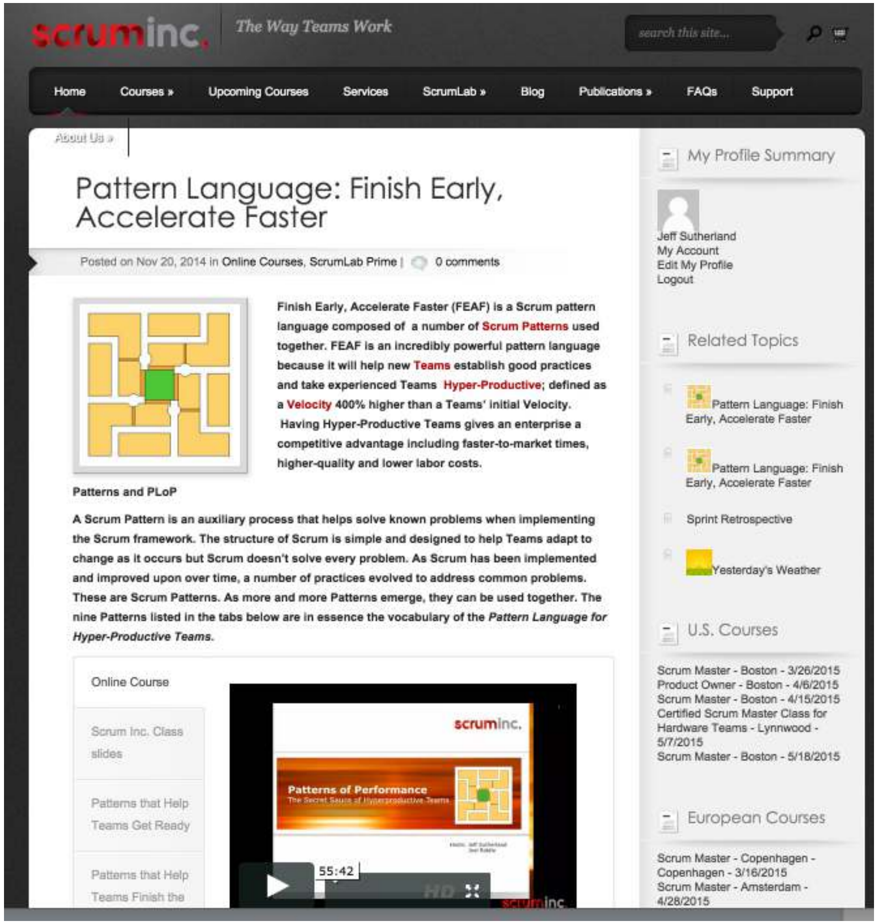
## Teams that Finish Early Accelerate Faster

- **Stable Teams** - How you get started
- **Yesterday's Weather** - How you pull backlog into a sprint
- **Daily Clean Code** - How to get defect-free product at sprint end
- **Swarming** - How you get work done quickly
- **Interrupt Pattern** - How to deal with interruptions during the sprint
- **Stop the Line** - How to deal with emergencies
- **Scrumming the Scrum** - How to ensure you improve continuously
- **Happiness metric** - How to ensure teams aren't overburdened

**Teams That Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams**
47th Hawaii International Conference on System Sciences (HICSS)
By Jeff Sutherland, Neil Harrison, Joel Riddle
January 2014

# ScrumLab Vision Statement

- **FOR** experienced Scrum practitioners (Jill) who are "in the trenches"
- **WHO need** clear and actionable information to answer specific Scrum questions whenever they arise
- **ScrumLab** is <u>the</u> authoritative, curated on-demand source for Scrum Inc.'s leading thinking
- **That:**
  - Clearly explains Scrum <u>and</u> its underlying principles (i.e. why if works)
  - Shares successful advanced practices for different contexts
  - Provides actionable solutions to implement successfully
  - Is available whenever you need it
- **Unlike** other online Scrum resources
- **Our product** captures decades of successful experience and wisdom from the co-creator of Scrum and his hand-picked team of thought leaders



Pattern Language: Finish Early, Accelerate Faster

Posted on Nov 20, 2014 in Online Courses, ScrumLab Prime | 0 comments

Finish Early, Accelerate Faster (FEAF) is a Scrum pattern language composed of a number of Scrum Patterns used together. FEAF is an incredibly powerful pattern language because it will help new Teams establish good practices and take experienced Teams Hyper-Productive; defined as a Velocity 400% higher than a Teams' initial Velocity. Having Hyper-Productive Teams gives an enterprise a competitive advantage including faster-to-market times, higher-quality and lower labor costs.

Patterns and PLoP

A Scrum Pattern is an auxiliary process that helps solve known problems when implementing the Scrum framework. The structure of Scrum is simple and designed to help Teams adapt to change as it occurs but Scrum doesn't solve every problem. As Scrum has been implemented and improved upon over time, a number of practices evolved to address common problems. These are Scrum Patterns. As more and more Patterns emerge, they can be used together. The nine Patterns listed in the tabs below are in essence the vocabulary of the *Pattern Language for Hyper-Productive Teams.*

Online Course

Scrum Inc. Class slides

Patterns that Help Teams Get Ready

Patterns that Help Teams Finish the

My Profile Summary

Jeff Sutherland
My Account
Edit My Profile
Logout

Related Topics

Pattern Language: Finish Early, Accelerate Faster

Pattern Language: Finish Early, Accelerate Faster

Sprint Retrospective

Yesterday's Weather

U.S. Courses

Scrum Master - Boston - 3/26/2015
Product Owner - Boston - 4/6/2015
Scrum Master - Boston - 4/15/2015
Certified Scrum Master Class for Hardware Teams - Lynnwood - 5/7/2015
Scrum Master - Boston - 5/18/2015

European Courses

Scrum Master - Copenhagen - Copenhagen - 3/16/2015
Scrum Master - Amsterdam - 4/28/2015

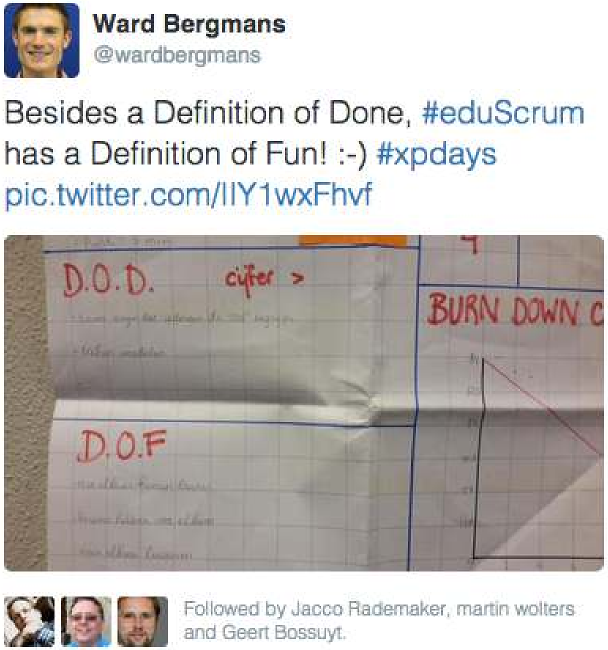
# Why Don’t Teams Have Working Software?

- Poor definition of DONE
- Stories not READY
- Dysfunctional leadership
- Technical debt
- Organizational debt
- Ineffective coaching

Source: ScrumInc/VersionOne Workshop 14 Oct 2014

# Poor Definition of DONE

- **Definition of DONE unclear**
  - It is impossible to be DONE if you don't know what DONE is.
- **Lack of consistent quality standards**
  - Definition of DONE does not include "working software".
  - Dysfunctional Product Owner accepts stories that are not done.



Ward Bergmans
@wardbergmans

Besides a Definition of Done, #eduScrum has a Definition of Fun! :-) #xpdays pic.twitter.com/lIY1wxFhvf

Followed by Jacco Rademaker, martin wolters and Geert Bossuyt.

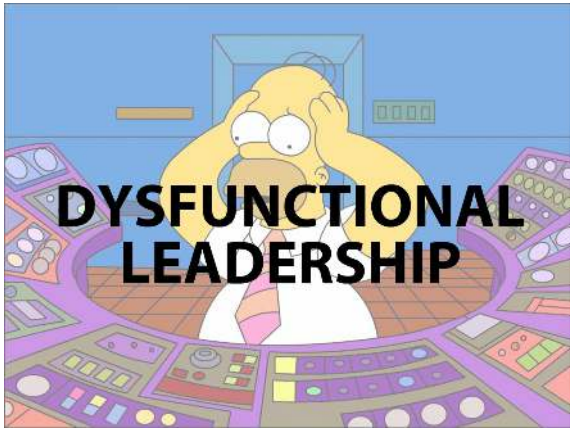
# Stories Not Ready

- **Sizing stories**
  - Bad estimates - inconsistent use of story points
  - Taking stories to big into sprint
  - Taking to many stories into sprint

- **Poorly written stories**
  - Stories not clear, particularly acceptance criteria
  - Unidentified dependencies

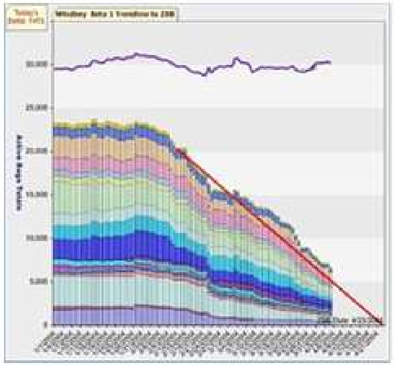
# Dysfunctional Leadership

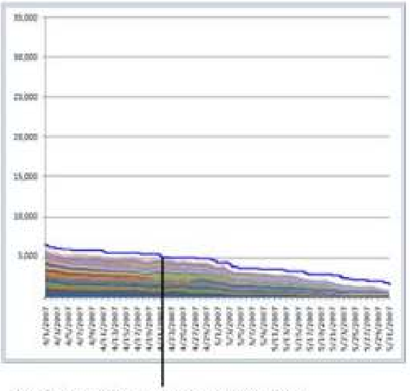
- Too many projects in pipeline (Context Switching)
- Everything is top priority
- Pressure to get things done delays projects and reduces quality
- Lack of understanding of Scrum
- Fear of exposure or change in responsibilities
- No continuous integration and/or no testing at all (Obamacare)



# Technical Debt

- Not finishing sprints creates bad code - 24x delay
- Legacy code is often accumulation of mountain of technical debt which reduces velocity
    - Severely aggravated by not using current technology for continuous integration and automated testing
    - Technical debt is incurred by running development too close to maximum which generates short cuts, lack of refactoring, loss of creativity, demotivation, and sloppy craftsmanship

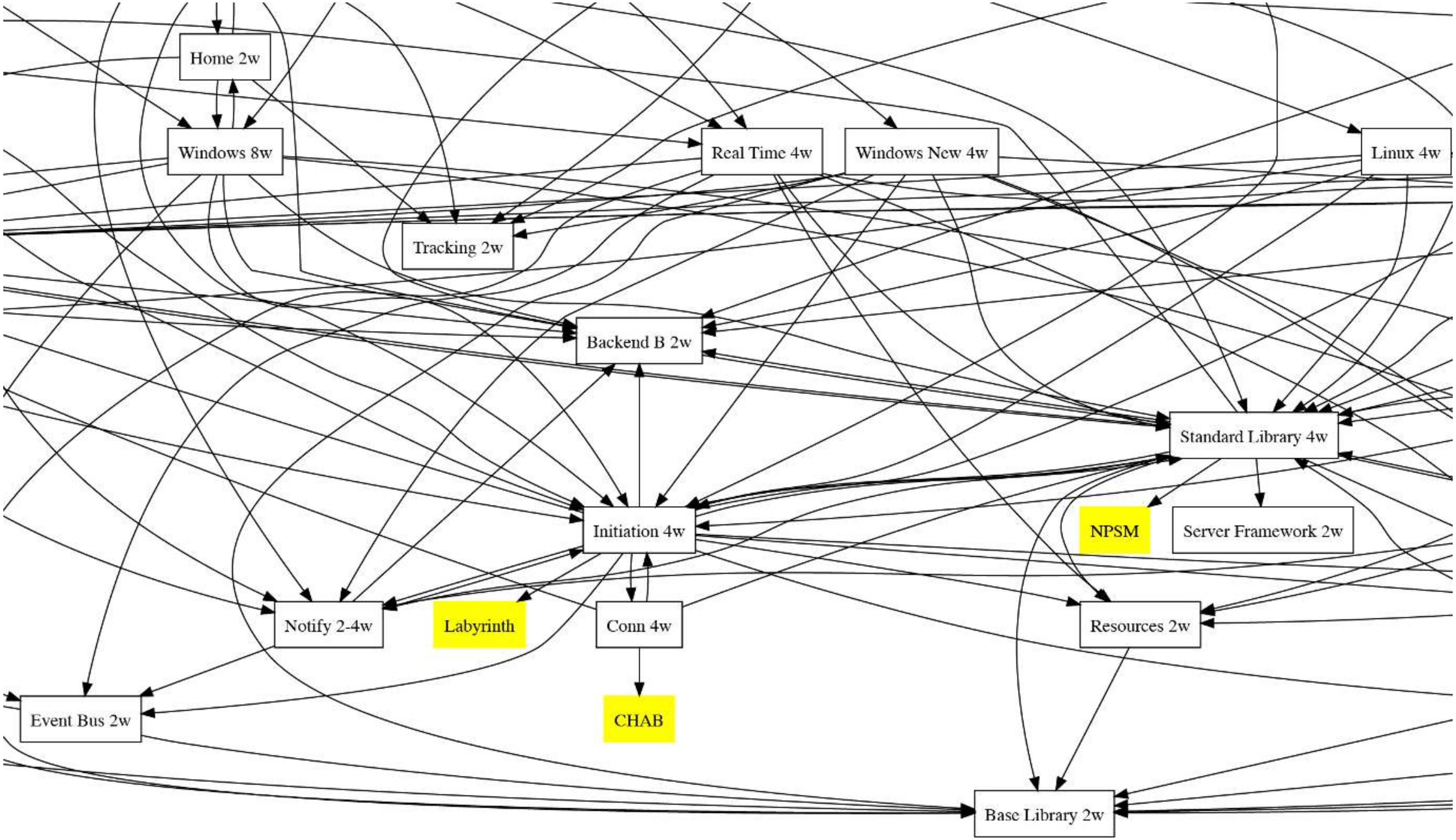




Total bug debt at Beta 1 of VS 2008.

Microsoft TFS Mountain of Technical Debt - Scrum reduced bugs from 30000 to 2000 - Agile Software Development with Vision Studio, 2011

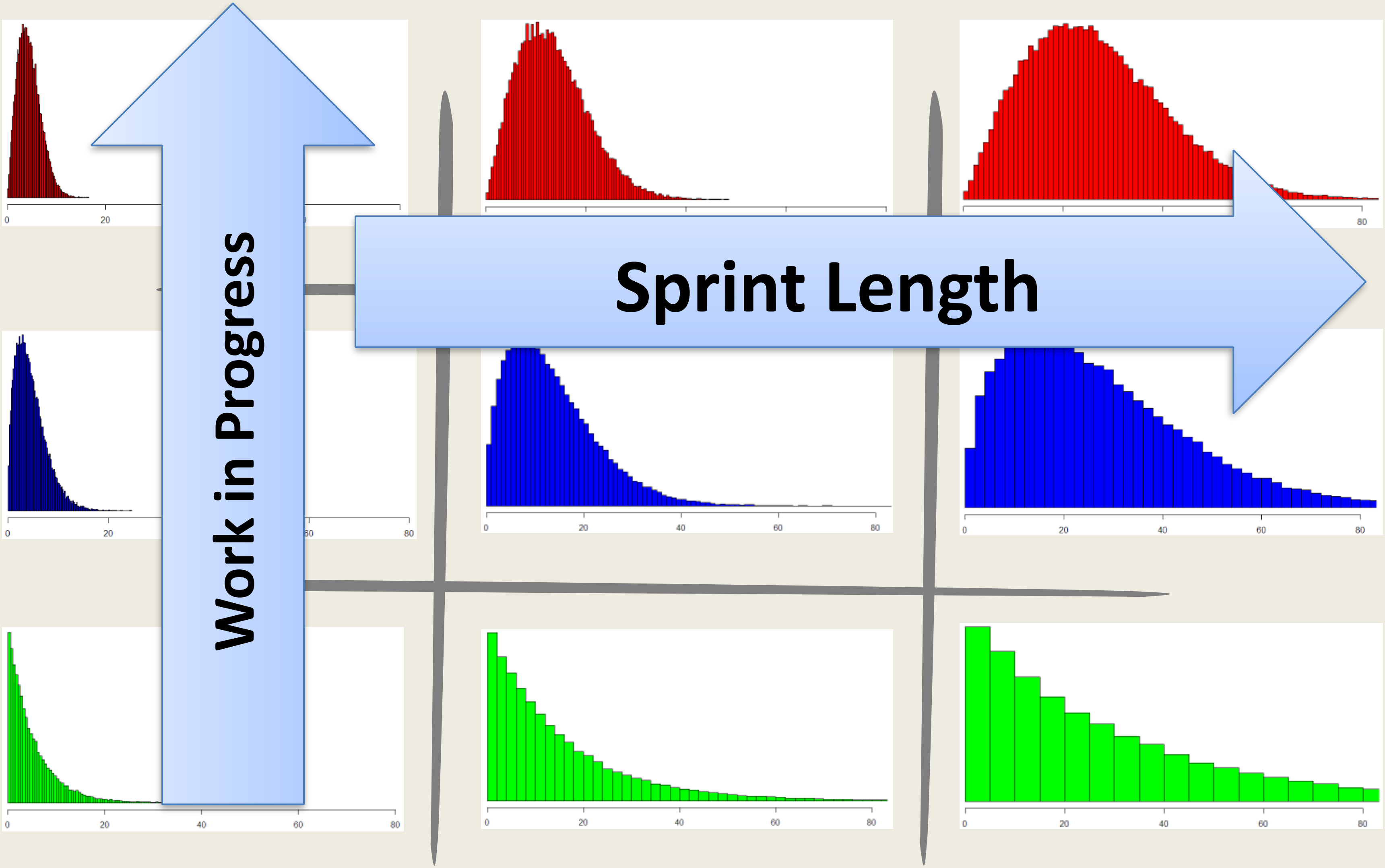
# Organizational Debt



**Agile Enterprise Metrics - 2015 48th Hawaii International Conference on System Sciences**
**Daniel R Greening, Senex Rex**
**dan@senexrex.com**

# Poor Coaching

- Silo’s and specialization cripple velocity
  - specialized test teams are the worst example
- Developers not functioning as a team
  - minimal collaboration, no swarming
- No continuous improvement flatlines velocity
  - no happiness
  - no interrupt pattern
  - no scrumming the scrum
- “Pretend Agile” - no teamwork, no working software, no customer collaboration, and no effective response to change



Work in Progress

Sprint Length

< 1 week

~ 2 week sprint

~ 1 month

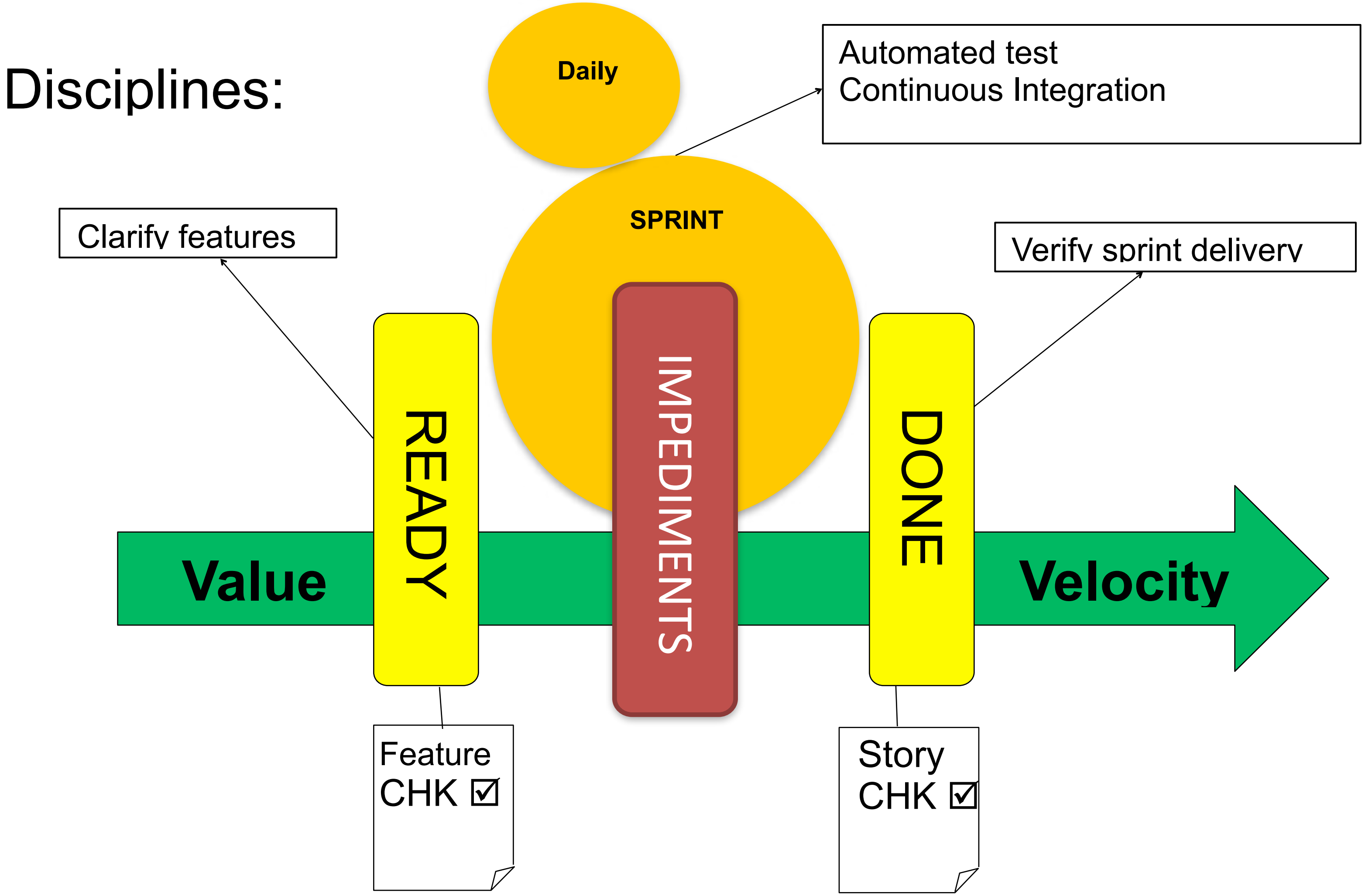
Time to Completion of a Story

Magennis, Troy. **The Economic Impact of Software Development Process Choice - Cycle-time Analysis and Monte Carlo Simulation Results.** 2015 48th Hawaii International Conference on System Sciences

# Systematic Approach to Getting To Done

- Implementing the Definition of Done
- Ensuring that backlog is Ready
- Training management
- Technical debt remediation plan
- Refactoring the organization
- Upgrading coaching and Scrum Master positions

# Systematic Scrum Model



Disciplines:

Daily

Automated test
Continuous Integration

SPRINT

Clarify features

Verify sprint delivery

READY

IMPEDIMENTS

DONE

Value

Velocity

Feature
CHK ☑

Story
CHK ☑

Scrum and CMMI - Going from Good to Great: Are You Ready Ready to Be Done Done
C. Jakobsen and J. Sutherland, in Agile 2009, Chicago, 2009.

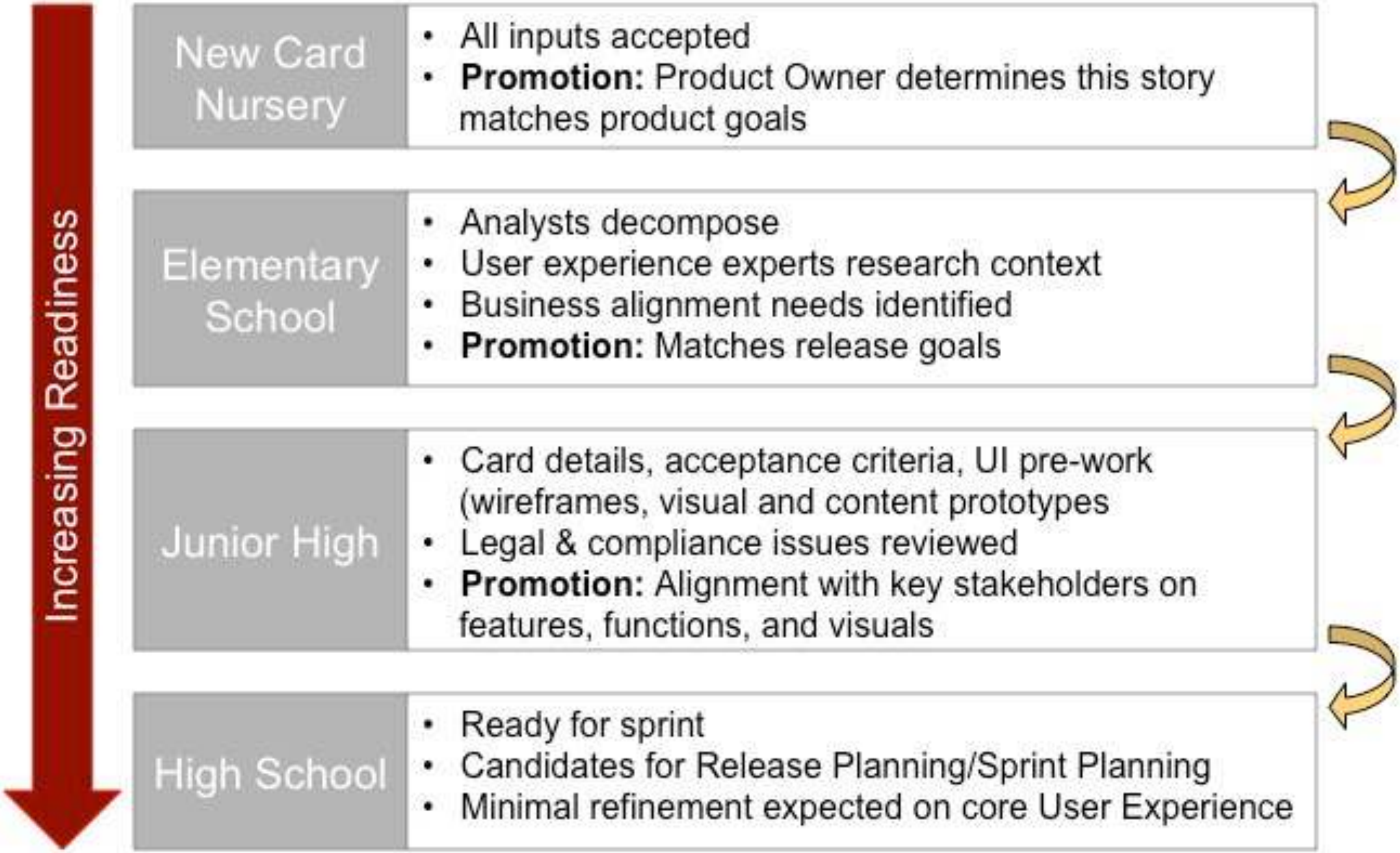
# Implementing Done

- **Definition of Done** must include integration testing and test capacity must exceed coding capacity
- **Testers must be on the Scrum team** - no separate test teams
- **Do not take too much into sprint**. Use Patterns.
  - Use “Yesterday’s Weather” pattern
  - “Illigitimus Non Interruptus”, and
  - “Scrum Emergency Procedure”
- **Use automated build system** combining new and old code (continuous integration)
- Systematically build **automated acceptance tests** which prevent top priority problems first
- Bugs **fixed in less than a day**
  - “Daily Clean Code”
  - 70% of defects are integration defects. Testing them later will take up to 24 times more testers!

# Implementing Ready

- Scrum Guide updated to include concept of **Ready**
- Team agrees on common **Definition of Ready**
- Only Ready Stories discussed at Sprint Planning
- **Backlog Refinement assures Ready** state.
- A good Ready state can **triple velocity**. Spend the time needed to get the backlog Ready.

# User Story Readiness Progression



**New Card Nursery**

- All inputs accepted
- **Promotion:** Product Owner determines this story matches product goals

**Elementary School**

- Analysts decompose
- User experience experts research context
- Business alignment needs identified
- **Promotion:** Matches release goals

**Junior High**

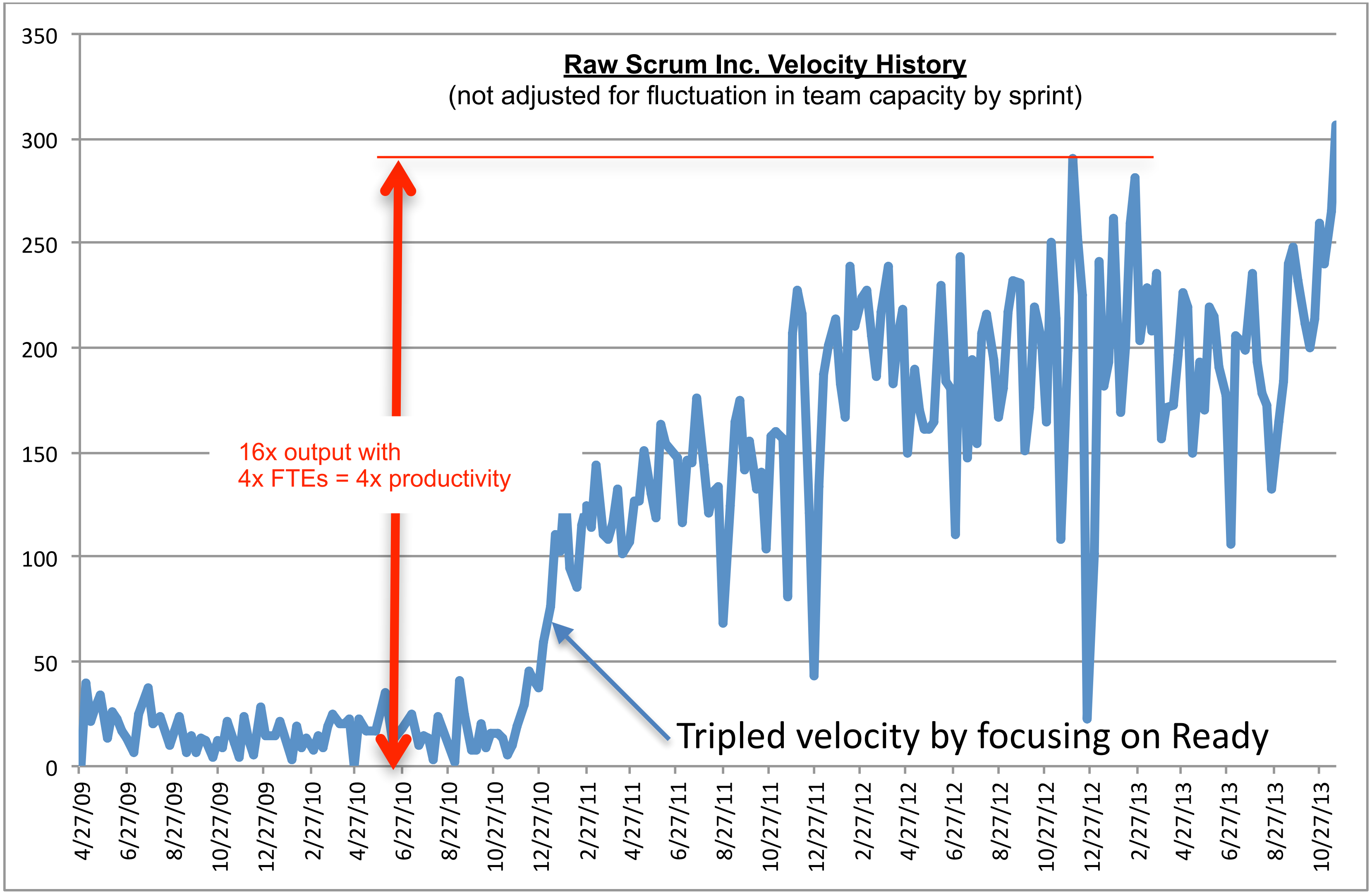
- Card details, acceptance criteria, UI pre-work (wireframes, visual and content prototypes
- Legal & compliance issues reviewed
- **Promotion:** Alignment with key stakeholders on features, functions, and visuals

**High School**

- Ready for sprint
- Candidates for Release Planning/Sprint Planning
- Minimal refinement expected on core User Experience

scruminc.

# Using Ready to Triple Velocity

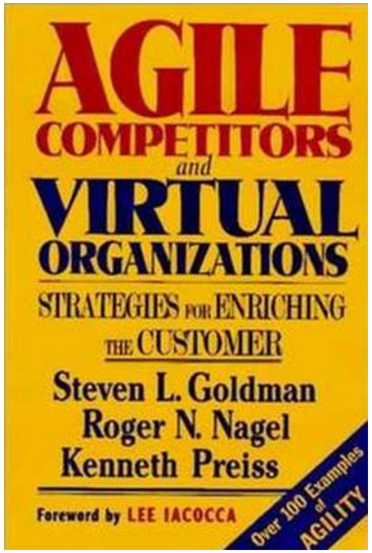


**Raw Scrum Inc. Velocity History**
(not adjusted for fluctuation in team capacity by sprint)

16x output with
4x FTEs = 4x productivity

Tripled velocity by focusing on Ready

# Functional Leadership



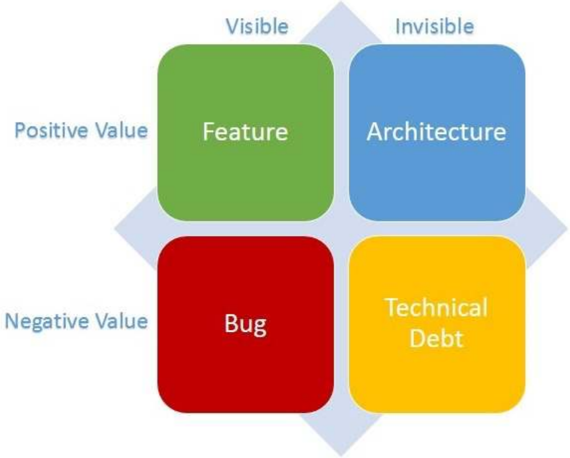
- **Agile competition goes beyond lean manufacturing** by permitting the customer, jointly with the vendor or provider, to determine what the product will be.
- For agile competitors, the ability to individualize products comes at little or no increase in manufacturing cost. It does, however exact a cost: **It requires major changes in organization, management philosophy, and operations.**
- Managers need to be trained in how to lead Agile teams.

# Leadership Responsibilities

- Provide challenging goals for the teams
- Create a business plan and operation that works
  - Set up the teams (in collaboration with teams)
  - Provide all resources the teams need
- Identify and remove impediments for the teams
  - Know velocity of teams
  - Understand what slows teams down (impediment list)
  - Remove waste (first-things-first)
- Hold P.O. accountable for value delivered per point
- Hold S.M. accountable for process improvement and team happiness

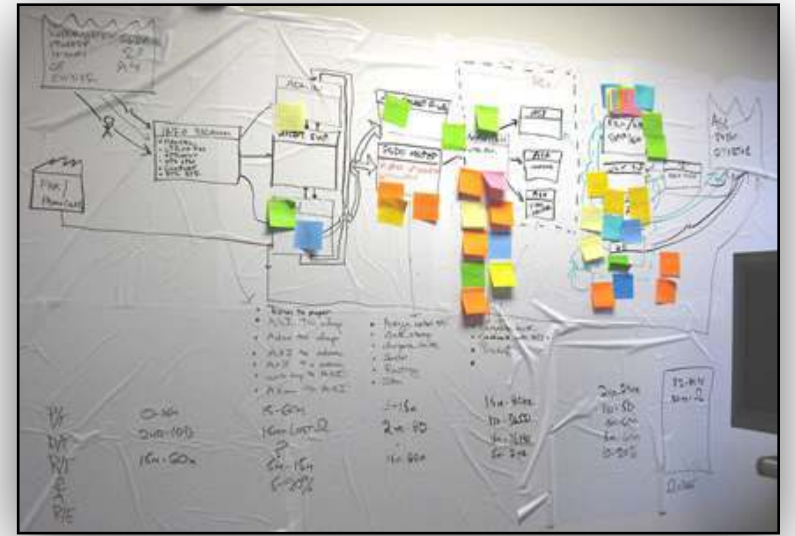
# Fix Technical Debt

- **Remediate**
  - 80% of bugs come from 20% of code (or less)
  - IBM’s strategy for determining remediation priorities - Mays et al. Experiences in Defect Prevention. IBM Systems Journal 29:1, 1990
- **Stop the Pain**
  - Systematically build acceptance tests into the build - highest priority first
- **Reduce the Debt**
  - Team build business case for Product Owner -
  - How many points for Tech Debt could could go to value creation? (How long will it take to remove debt?)
- **Management commits to systematic improvement of product**
  - Reduce operational costs
  - Increase sales



# Eliminate Organizational Debt

- Identify blocking teams and fix them
- Move towards cross-functional feature teams
- Train and hire T-shaped people
- Build out an object-oriented architecture for the product
- Use value stream mapping to identify queues, wait states, and other waste across the organization



# Spotify Succeeds with Excellent Coaching

- Hires great workers
- Every team has a coach
  - Coaches are responsible for 1 process improvement every Sprint
  - Improvement backlog and they measure improvement continuously
  - Coaching has radically improved output of high performance teams.
- In the last year, 33% of all Spotify Teams have moved to continuous deployment multiple times per sprint.

# Cycle Time Improvement

## When Velocity Goes Up Cycle Time Goes Down

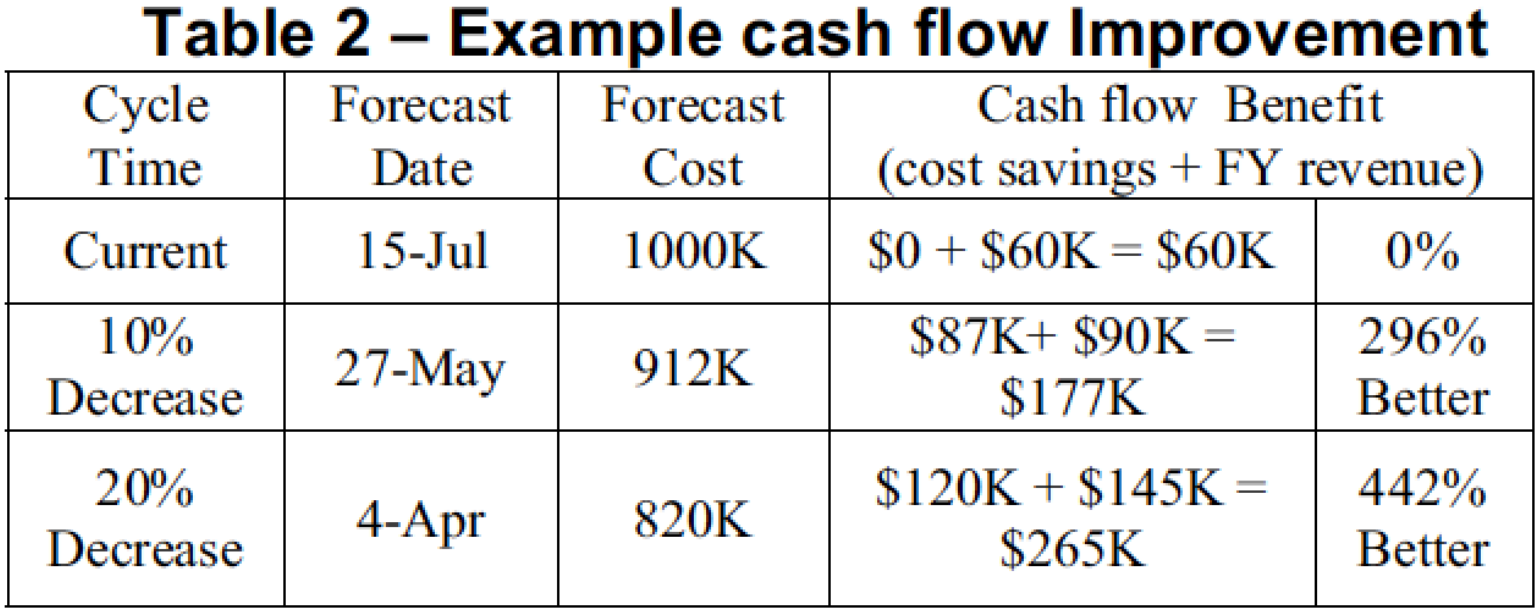


Table 2 – Example cash flow Improvement

| Cycle Time | Forecast Date | Forecast Cost | Cash flow Benefit (cost savings + FY revenue) | |
|---|---|---|---|---|
| Current | 15-Jul | 1000K | $0 + $60K = $60K | 0% |
| 10% Decrease | 27-May | 912K | $87K+ $90K = $177K | 296% Better |
| 20% Decrease | 4-Apr | 820K | $120K + $145K = $265K | 442% Better |

Magennis, Troy. **The Economic Impact of Software Development Process Choice - Cycle-time Analysis and Monte Carlo Simulation Results.** 2015 48th Hawaii International Conference on System Sciences

# Best Metrics for Coaches

- **Time to fix a defect**. If this averages less than 24 hours the team's velocity will double.
- **Measure of swarming.** How well do individuals and interactions generate performance.
  - Measure flow = actual work to do a story/ calendar time to done
  - If this is over 50% team velocity will double again

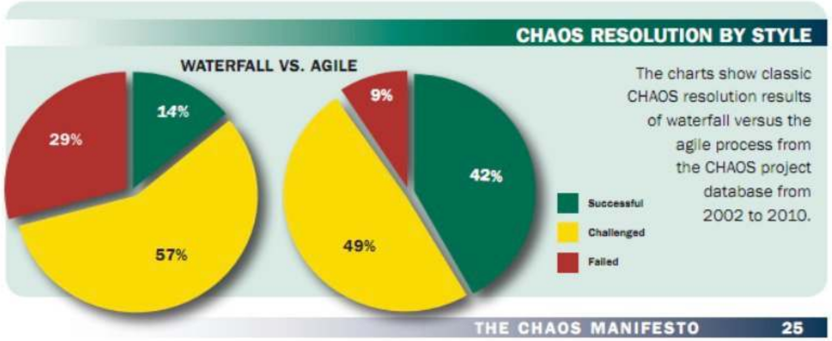
**Going from Good to Great with Scrum**
**Are you READY READY to be DONE DONE?**

Carsten Ruseng Jakobsen and Jeff Sutherland

SYSTEMATIC scruminc.

# Conclusions

- Bad Agile is caused by five primary factors
  - Poor definition of DONE
  - Stories not READY
  - Dysfunctional leadership
  - Technical debt
  - Organizational debt
  - Ineffective coaching
- Systematically focusing on remediating these issues will produce high performing teams with 200-400% improvement in production.
- Failure to focus on them will add yet another team to the 58% of teams that are “Bad Agile” leading to unhappy customers, lost revenue, and lower stock prices.



CHAOS RESOLUTION BY STYLE

The charts show classic CHAOS resolution results of waterfall versus the agile process from the CHAOS project database from 2002 to 2010.

# Questions?

Wallabies

# Stay Connected

## Scruminc.com

- For up coming events, new content releases, and more!

## ScrumLab

- articles, online courses, tools, and papers on all things scrum
- www.scruminc.com/scrumlab

## Blog

- http://www.scruminc.com/category/blog/

## Online Courses

- advance your scrum with our online courses. Visit the Scrumlab store to view upcoming topics.

## Twitter, Facebook, and G+

- @ScrumInc, @jeffsutherland, scrum and scrum inc.