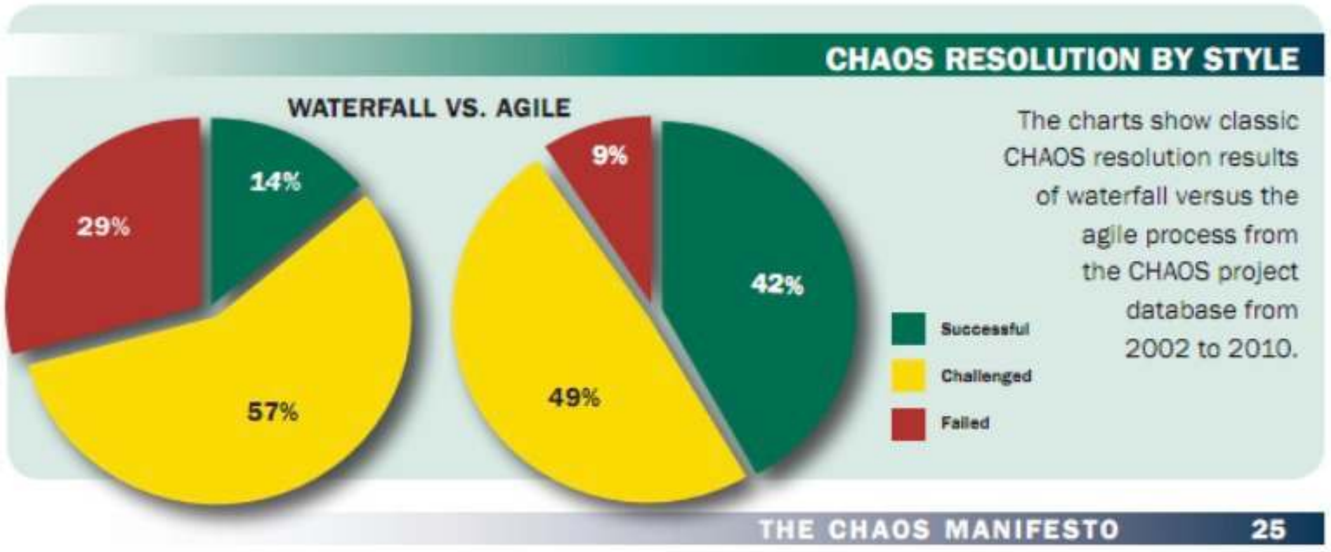
scruminc.

Getting to Done
The Secret Sauce of High Performing Teams

Hosts: *Lowell Lindstrom*
Jeff Sutherland

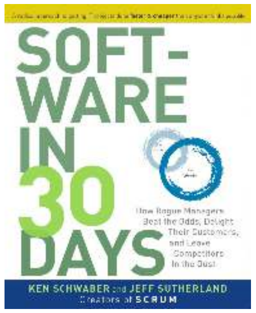
Is Your Project Agile?



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

49% of Agile is “Bad Agile”

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.

What is the Primary Reason for “Bad Agile”?

- **Failure to implement the Agile Manifesto**.
  - *Individuals and Interactions over Processes and Tools*
  - *Working Software over Comprehensive Documentation*
  - *Customer Collaboration over Contract Negotiation*
  - *Responding to Change over Following a Plan*
- **Teams do not work together** to produce working software at the end of a sprint!
- Teams cannot respond to stakeholder feedback at the end of a sprint because the software doesn’t work!
- Fixing bugs later can mean 24 x more testing!

Wannabe Agile

scruminc.

Why Is It So Important to Have Working Software?

- ScrumInc provides agile coaching to Openview Venture Partners (since 2006). All employees are on Scrum teams and all portfolio companies do Scrum.
- After running thousands of sprints, OpenView investors did a detailed analysis of data in Scrum tooling and discovered:

Teams That Finish Early Accelerate Faster!

scruminc.

Why Don't Teams Have Working Software

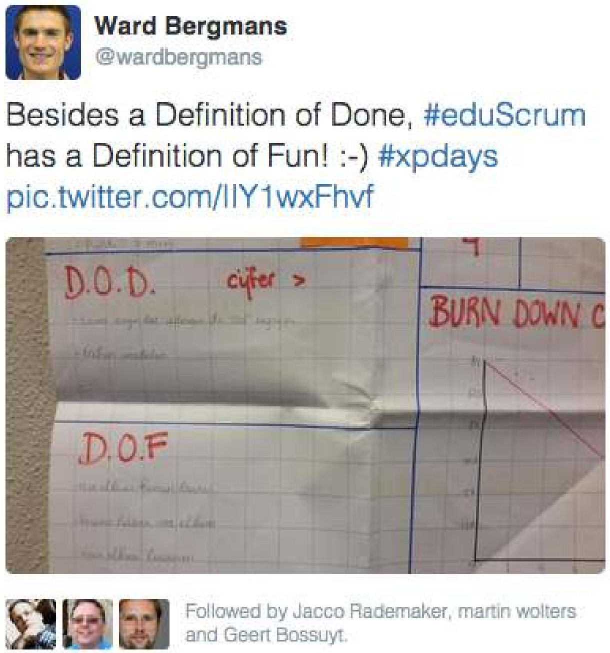
- Poor definition of DONE
- Stories not READY
- Dysfunctional leadership
- Technical debt
- Ineffective coaching

Source: ScrumInc/VersionOne Workshop 14 Oct 2014

scruminc.

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.

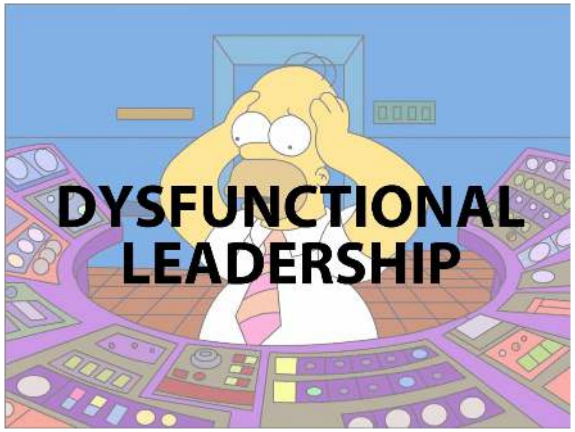
Stories Not Ready

- **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

scruminc.

Dysfunctional Leadership

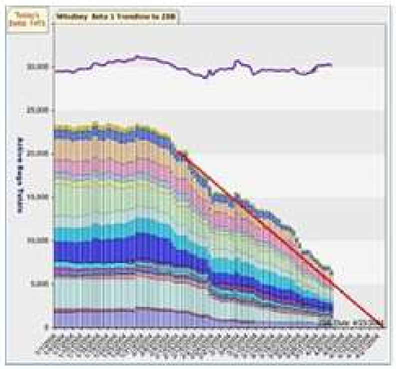
- Too many projects in pipeline (Context Switching)
- Everything is top priority
- Pressure to get things done delays projects and reduces quality
- Lack of understanding of Scrum
- Fear of exposure or change in responsibilities
- No continuous integration and/or no testing at all (Obamacare)

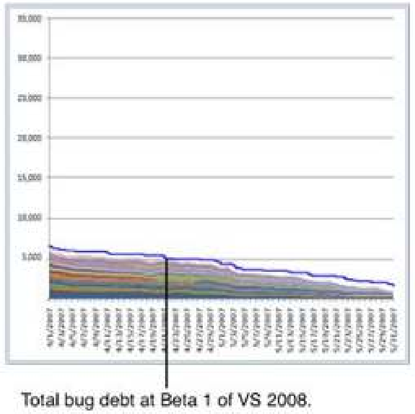


scruminc.

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





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

scruminc.

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

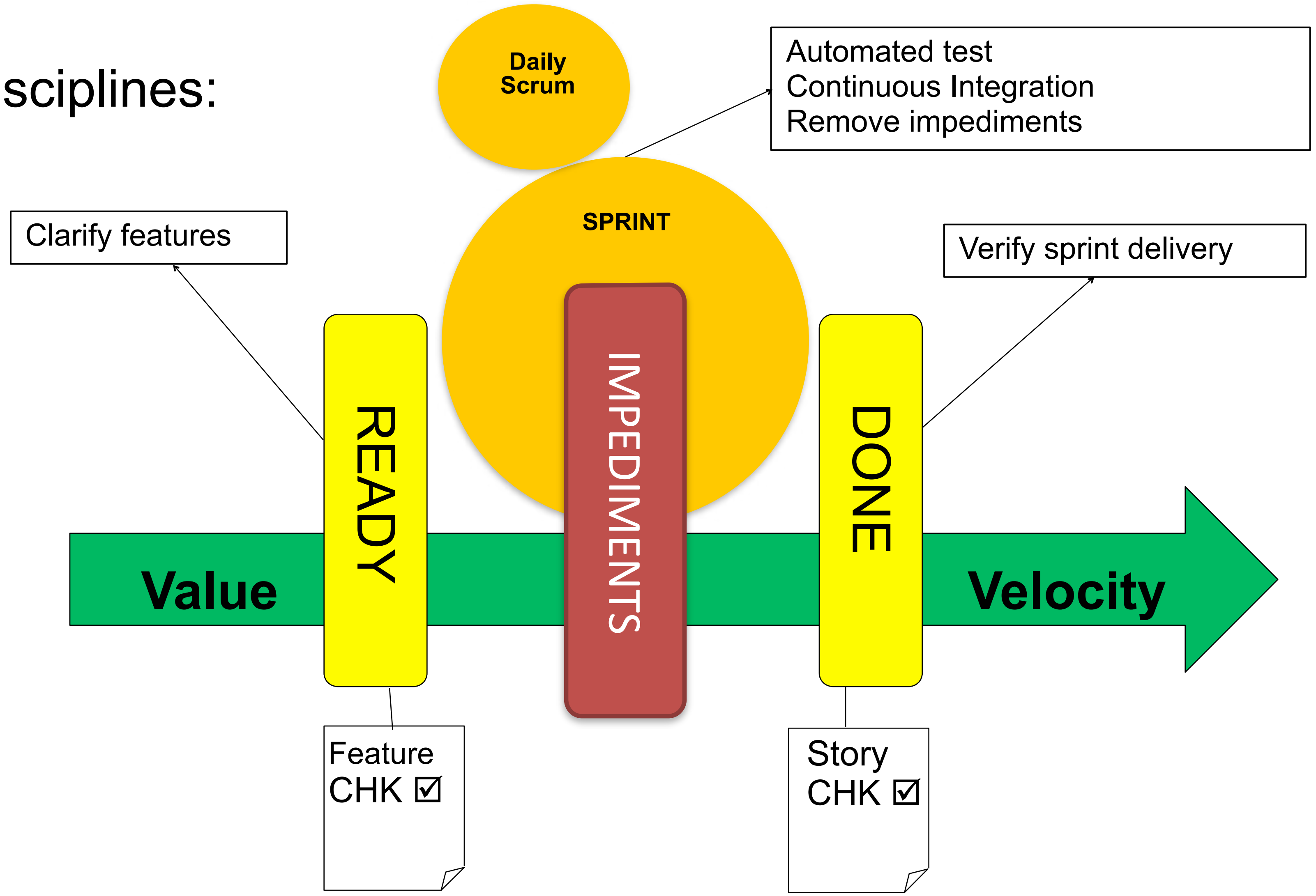
Systematic Approach to Getting To Done

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

scruminc.

Systematic Scrum Model

Disciplines:



scruminc.

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 into Sprint Backlog
- **Backlog Refinement assures Ready** state.
- A good Ready state can **triple velocity**. Spend the time needed to get the backlog Ready.

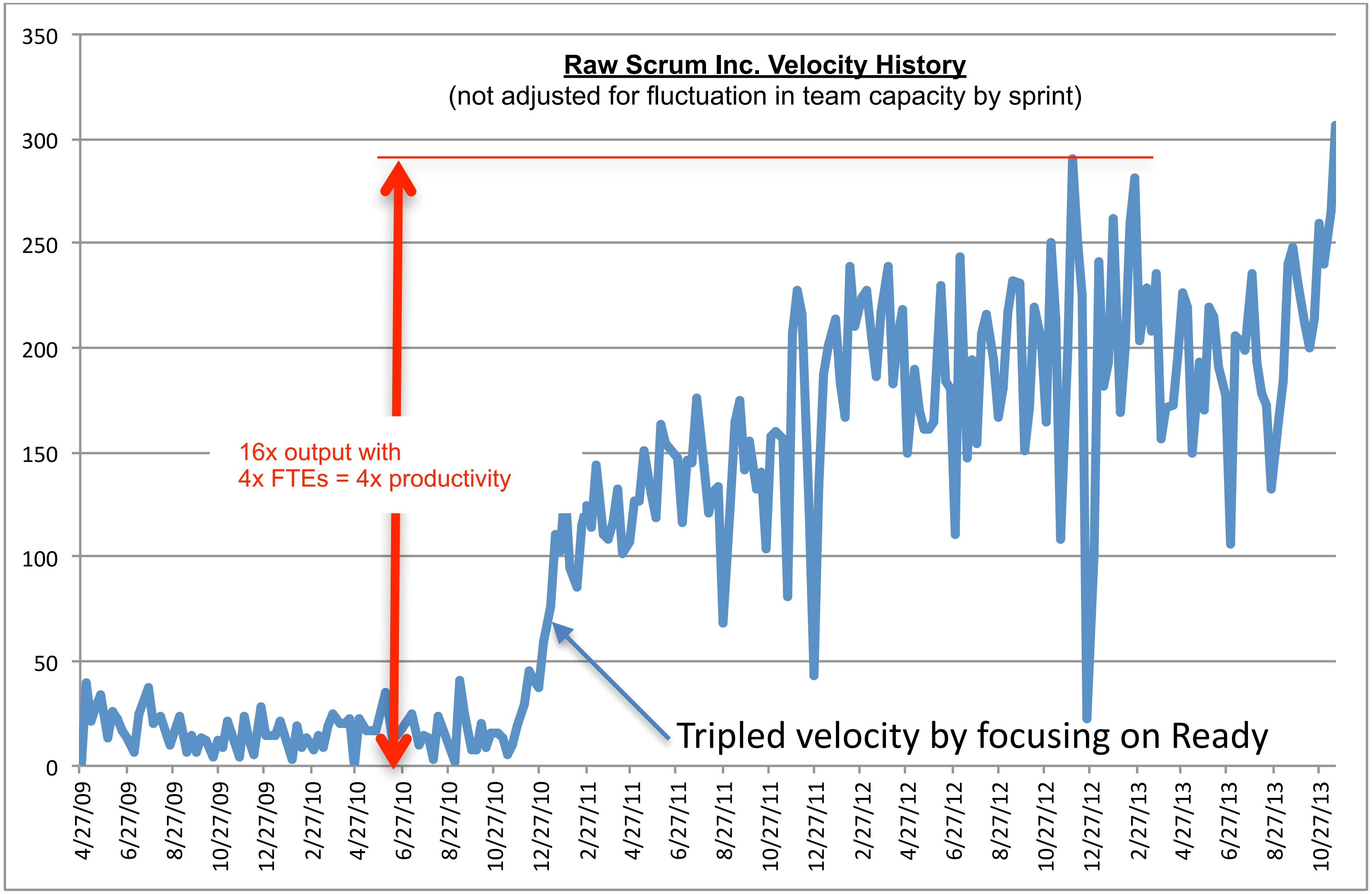
scruminc.

User Story Readiness Progression

Increasing Readiness

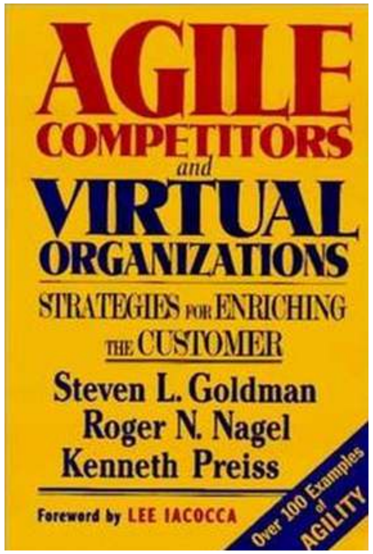
New Card Nursery	• All inputs accepted • **Promotion:** Product Owner determines this story matches product goals
Elementary School	• Analysts decompose • User experience experts research context • Business alignment needs identified • **Promotion:** Matches release goals
Junior High	• Card details, 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

Using Ready to Triple Velocity



scruminc.

Functional Leadership



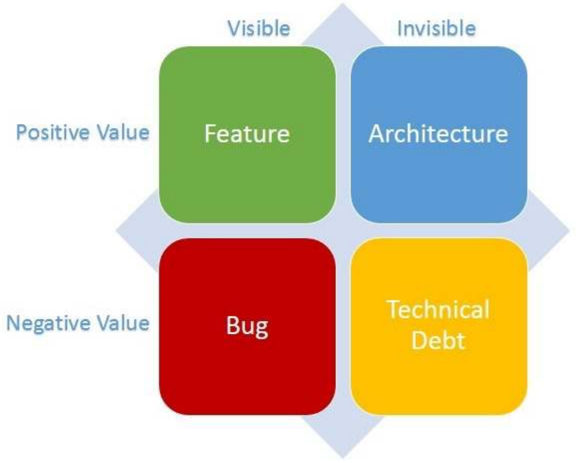
- **Agile competition goes beyond lean manufacturing** by permitting the customer, jointly with the vendor or provider, to determine what the product will be.
- For agile competitors, the ability to individualize products comes at little or no increase in manufacturing cost. It does, however exact a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

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



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.

scruminc.

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.

scruminc.

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

Conclusions

- Bad Agile is caused by five primary factors
  - Poor definition of DONE
  - Stories not READY
  - Dysfunctional leadership
  - Technical debt
  - Ineffective coaching
- Systematically focusing on remediating these issues will consistantly produce high performing teams with 200-400% improvement in production.
- Failure to focus on them will add yet another team to the 49% of teams that are “Bad Agile” leading to unhappy customers, lost revenue, and lower stock prices.

Questions?

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.