

Scrum Pitfalls (I)

And How to Navigate them Safely



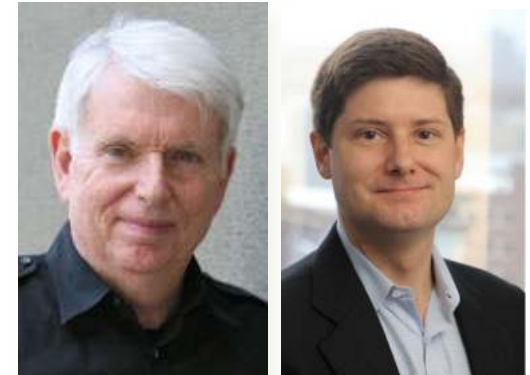
Host: Alex Brown
Presenter: Jeff Sutherland

scruminc.: Who We Are

Scrum Inc. is the Agile leadership company of Dr. Jeff Sutherland, co-creator of Scrum. We are based in Cambridge, MA.

We maintain the Scrum methodology by:

- Capturing and codifying evolving best practices,
- Conducting original research on organizational behavior
- Adapting the methodology to an ever-expanding set of industries, processes and business challenges



We also help companies achieve the full benefits of Scrum through our full suite of support services:

- Training (Scrum Master, Product Owner, Agile Leadership, webinars, etc.)
- Consulting (linking Scrum and business strategy, customizing Scrum)
- Coaching (hands-on support to Scrum teams)
- Publishing and new content development

We run our services company using Scrum as the primary management framework, making us a living laboratory on the cutting edge of “Enterprise Scrum”

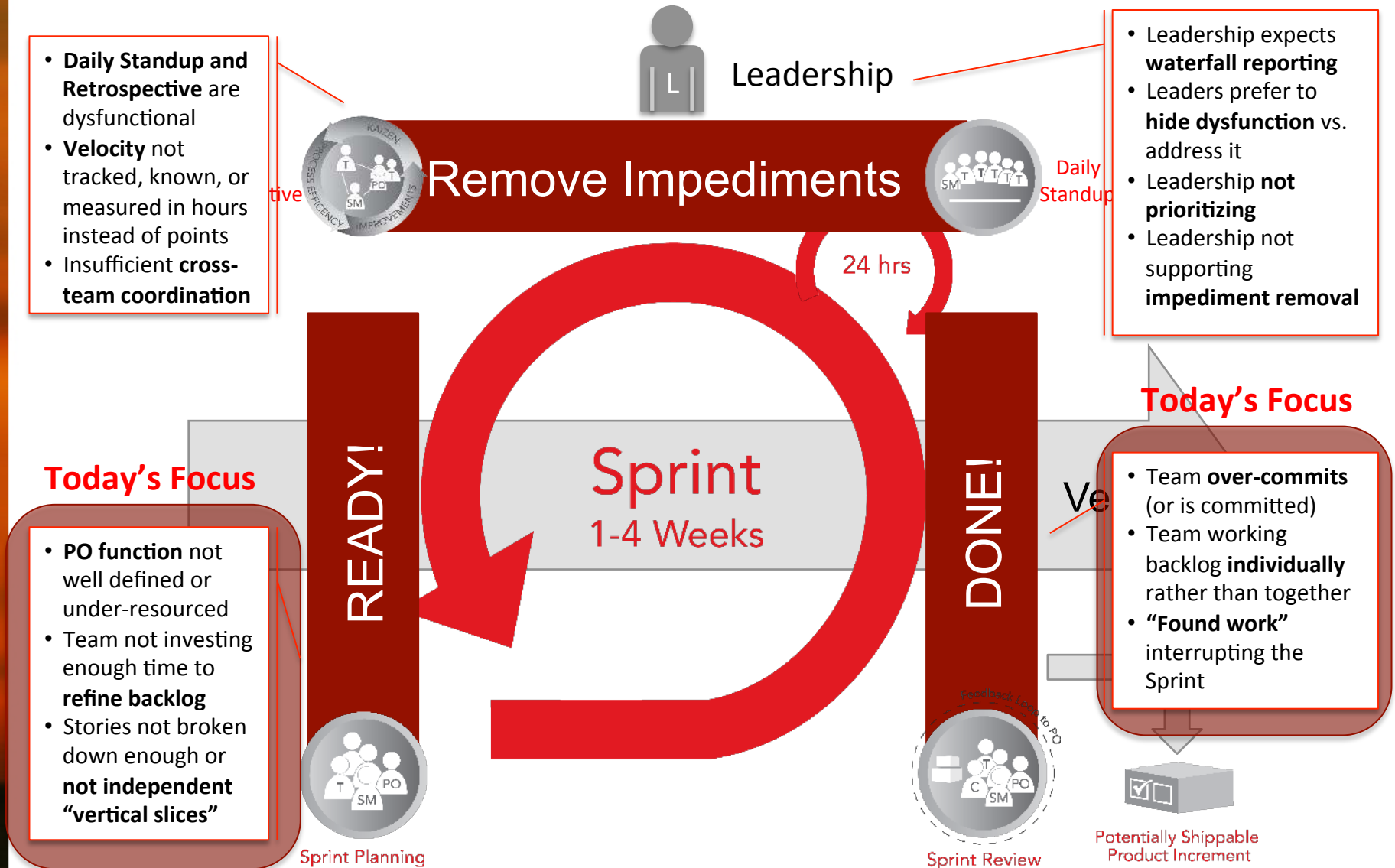
Find out more at www.scruminc.com.

Agenda

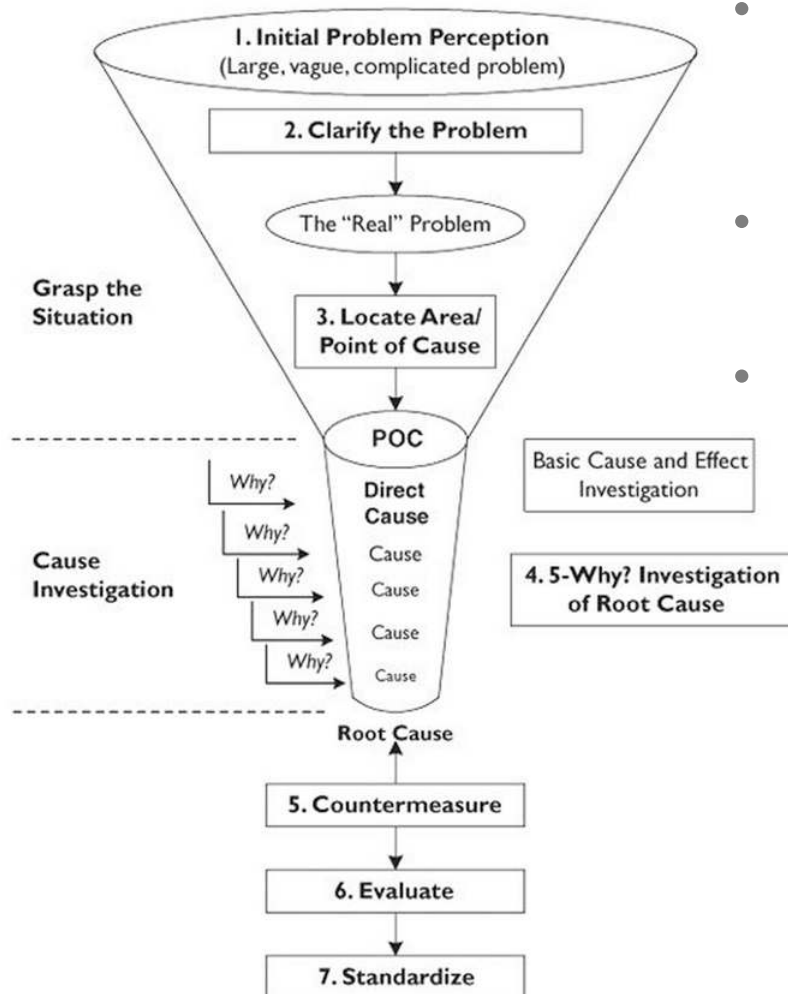
- Introduce model for Scrum effectiveness and associated pitfalls
- Discuss the A3 Process as a tool for identifying and overcoming typical pitfalls
- Review the 7 most common Scrum pitfalls related to READY and DONE
- Q&A

A Simple Model for How Scrum Works

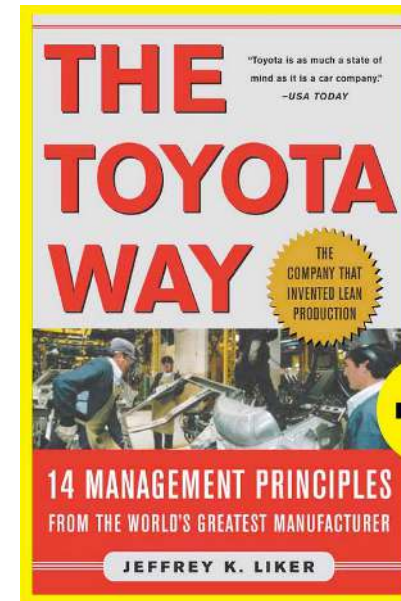
And the Pitfalls that Can Cause it to Break Down



Important to Address Root Causes Rather than Just Treat Symptoms



- The A3 is a light-weight problem-solving tool designed to identify and address root causes
- Came out of Toyota's continuous improvement process
- Named for size of paper (11x17)



Venture Company Example

A3 Process Creates Pull-Based Authority

Title: Seven teams failing too many sprints	
<u>Background</u> <ul style="list-style-type: none">•Teams not getting software done and tested•Critical components failing every other sprint P	Owner: Mentor: Date:
<u>Current Condition</u> <ul style="list-style-type: none">•Engineers not working together?•Inability to test causing failure•Waste estimated at 2.1M Euro/year L	<u>Countermeasures (Experiments)</u> <ul style="list-style-type: none">•Meet with board member•Conference call with CEO•Commitment to implement continuous integration•Site visit to demonstrate working processes Do
<u>Goal / Target Condition</u> <ul style="list-style-type: none">•Clean tested code worked at end of sprint•Cut waste by 90%•Save 1.8M Euro/year while improving quality A	<u>Confirmation (Results)</u> <ul style="list-style-type: none">•Clean implementation in one month•Velocity of seven teams average increase of 20%•Immediately savings of 1.7M Euro/year•Cost of implementation 3000 Euro for expert consultant Check
<u>Root Cause Analysis</u> <ul style="list-style-type: none">• Why- engineers had different design concepts•Why- Team members not communicating•Why- ScrumMaster not doing good job•Why- No continuous integration•Why- Product Owner focused on new features N	<u>Follow-up (Actions)</u> <ul style="list-style-type: none">•Introduced prioritized automated testing•Introduced code reviews•Cut deployment time in half•Cut support calls in half•Increased sales Act

PO Role Not Defined or Under-Resourced

Impediments

Ready



Done

Typical symptoms

- Stories frequently not done at Review due to external dependencies or in-sprint surprises
- Product Owner not available to answer Team questions in a timely fashion
- Many stories “discovered” during the Sprint
- Team feels priorities shifting too frequently
- Team gets conflicting messages from different sources

Root causes

- User stories not clear and READY at start of Sprint
 - Needed information not available in time
 - Poor clarity on who is responsible for providing what information
 - Unclear who leads story creation/refinement
 - **Product Owner role is not well-defined**
 - Single PO creating all backlog for multiple teams or all customer engagement thru to story creation for one accelerating team
 - **Product Owner role under-resourced**
- Conflicting Team goals from multiple sources
 - Unresolved competing stakeholder interests
 - **Product Owner role is not well-defined**

What to do about it

PO role not defined

- Assemble all stakeholders to decide on the single tactical PO to work with team
- All backlog should flow to team through PO
- Set up regular Meta-Scrum meeting for stakeholders to align without impacting team

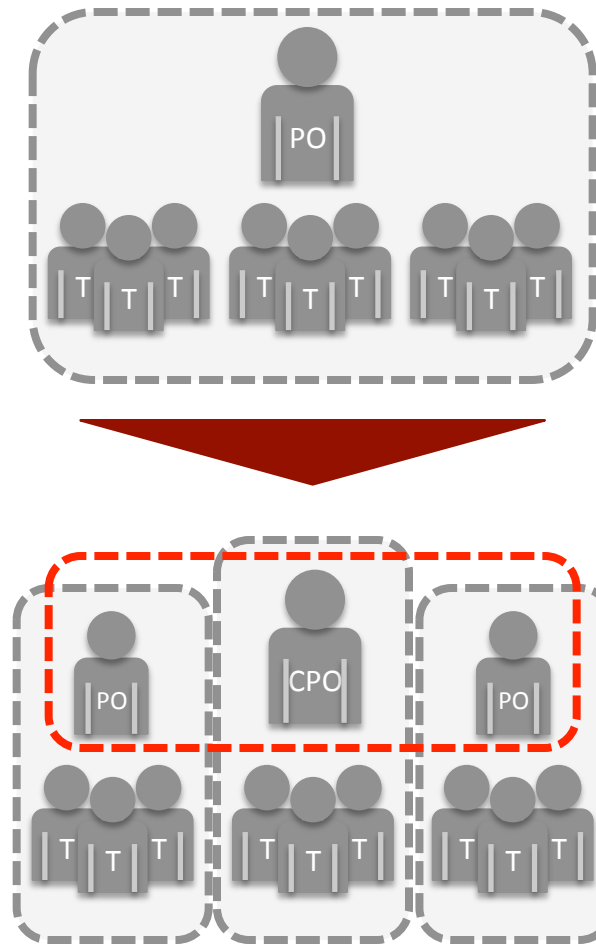
PO role under-resourced

- Ensure that each team has its own PO
- Designate separate Strategic (epic-level market and ROI) and Tactical (ready backlog) PO to work closely together
- Assign cross-functional PO team

Target end-state

- Stakeholders have an aligned and compelling vision that is maintained regularly
- This vision communicated to Team through the PO and a consistent Product Backlog
- Backlog stories follow regular refinement process to ensure they are ready before Sprint Planning
- Progress communicated back to stakeholders without distracting the Team

Product Owner is a Big Job



- Initially, one Product Owner may be able to generate ready backlog for several teams
- As team velocity increases, a Product Owner team, led by a Chief Product Owner, will be needed
- The Product Owner team are domain experts that describe the user experience, the screen shots, the workflow, the data requirements, the look and feel.

Stories Aren't Ready Before Sprint Planning

Impediments

Ready



Done

Typical symptoms

- Sprint Planning Meeting is tedious and takes a long time to complete, maybe even a full day
- Team has many questions during Sprint Planning that PO cannot answer during the meeting
- Stories are difficult to estimate at Sprint Planning
- At the end of each Sprint there are several stories not finished or not even started

Root causes

- Team writing lots of new stories at Planning
 - New stories needed to deliver Sprint priorities
 - Team sees upcoming work for the first time
 - **Team not investing in Refinement**
- Lots of unplanned work emerges during the Sprint
 - Research or clarification often required to begin work planned
 - Team hasn't thought all work needed to deliver the story
 - **Team not investing in Refinement**
- PO needs input from external stakeholders
 - Team needs more information to plan
 - PO hadn't anticipated required lead time
 - **Team not investing in Refinement**

What to do about it

SM encourage Team to look ahead

- Adopt mindset of looking forward to anticipate questions, dependencies and risks
- Coordinate regular Refinement meetings for Team and PO to discuss future sprints
- Coach team to utilize INVEST criteria

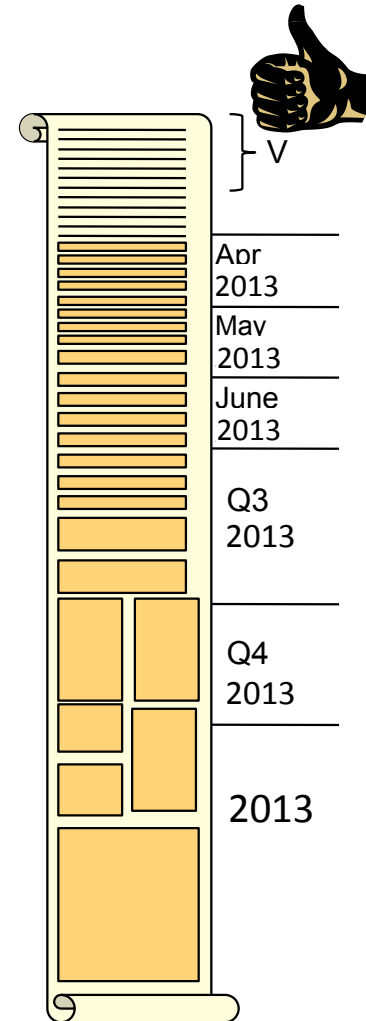
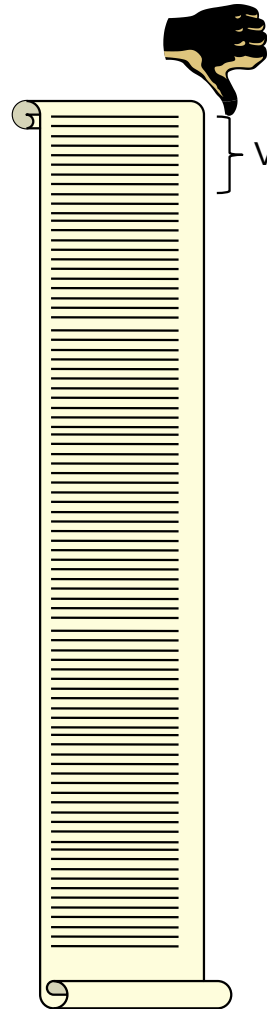
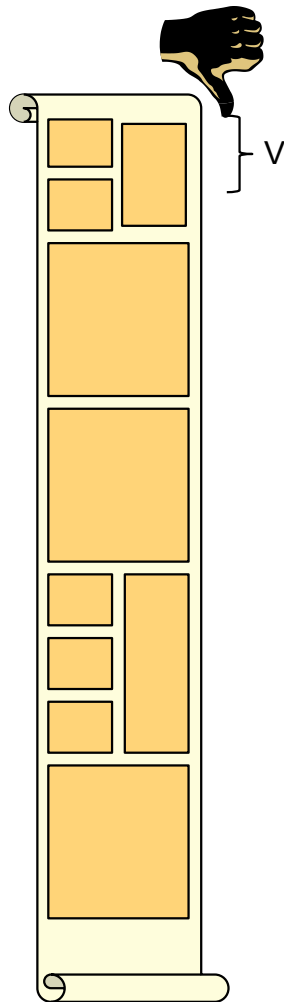
PO meet with Team before each Sprint

- Approach specific Team members with questions needed to prepare Sprint Backlog
- Attend Refinement meetings with Team to explain upcoming work, get Team clarification
- Clarify work with stakeholders before Planning

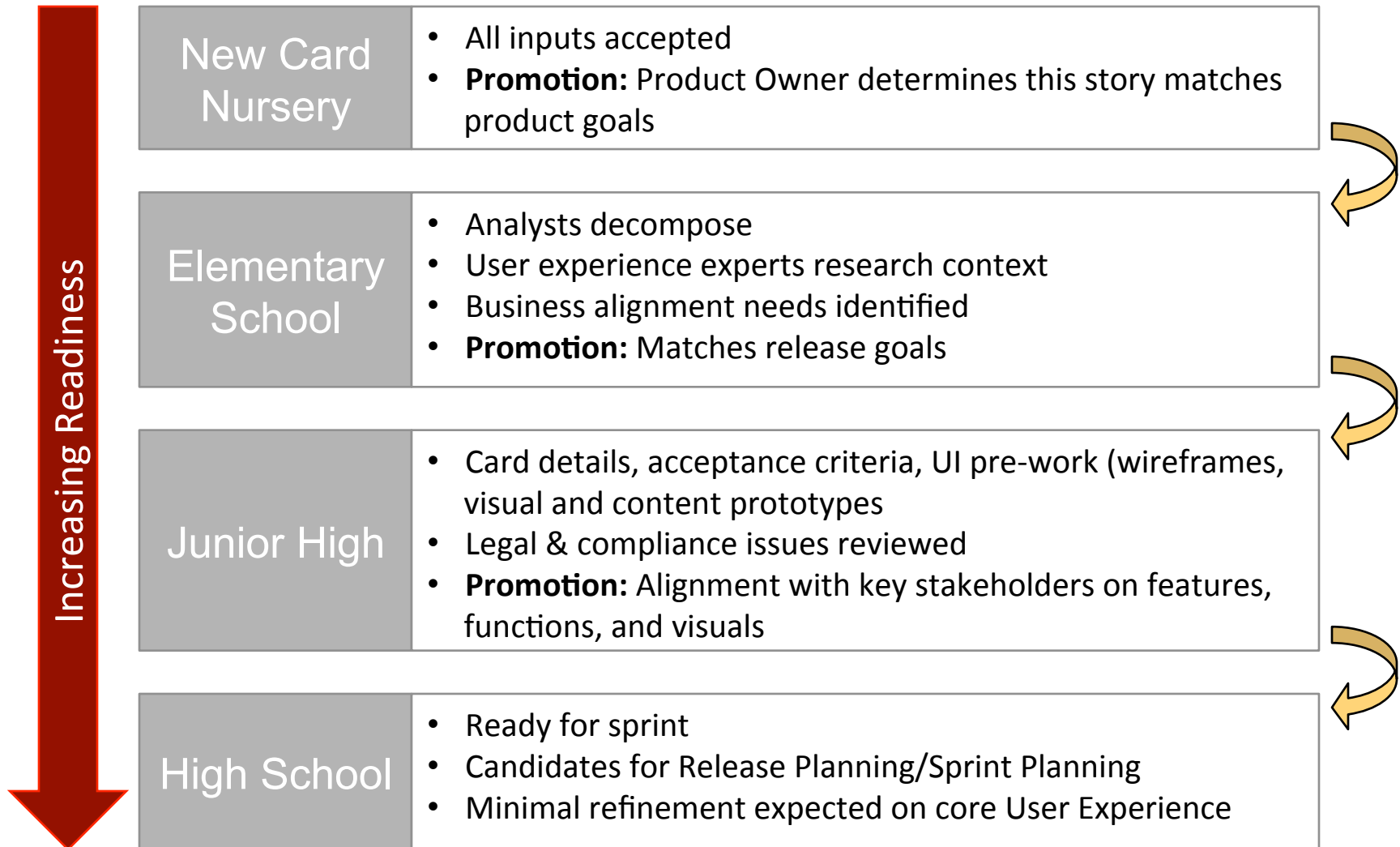
Target end-state

- Shorter and more effective Sprint Planning meetings (1 hour or less per week of Sprint)
- Few "surprises" during Sprint that could have been avoided with better planning
- Team finishes planned work ~80%+ of Sprints
- Team and PO work together to Refine backlog (expect 5-10% of the Team's time)

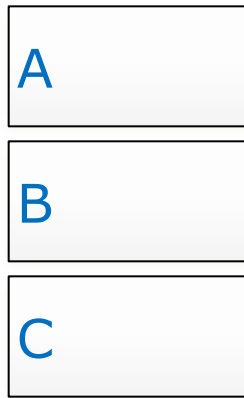
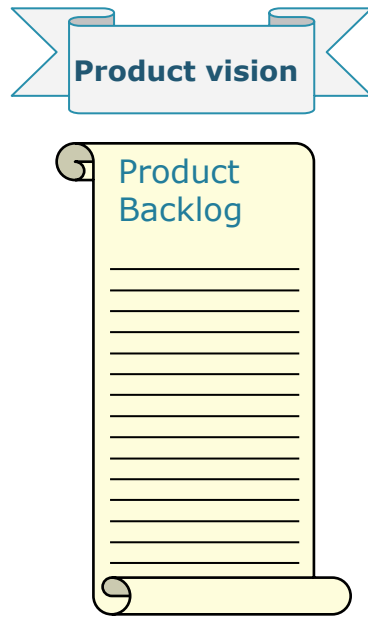
Team Works to Maintain the Right Progression of Backlog Definition



User Story Readiness Progression

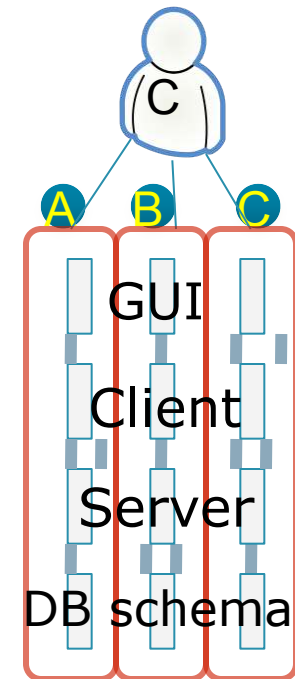


User Story Readiness Guidelines



Immediately actionable
Negotiable
Valuable
Estimable
Sized to fit
Testable

Modified from Bill Wake – www.xp123.com



PBIs Not Broken Down into Small Vertical Slices of Functionality

Impediments

Ready



Done

Typical symptoms

- Stories usually involve only one discipline or team member (function-centered stories)
- Stories difficult for team to act on immediately
- Several stories must be completed before functionality would create value for customers
- Multiple days pass without team completing a story (uneven burndown)
- Actual work often much greater than estimated

Root causes

- Team struggles to work together on PBIs
 - PBI definition includes only one person/functionality from team
 - **Defined from team not user perspective**
- Multiple stories must be completed before incremental functionality ships
 - **PBIs address only one functional element**
- Actual work often much greater than estimated
 - Not all team members participate in estimation for function-centered PBIs
 - Team members think “it isn’t my work”
 - **PBI not defined as vertical functionality**

What to do about it

PBI's Not Defined As Vertical Slices of Functionality

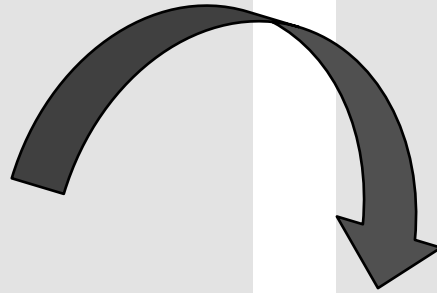
- Make sure every PBI is in “User Story” form, or at least Team can identify how PBI generates incremental customer value
- Get entire team to agree on clear Definition of Ready for all Backlog items that aligns with target end state
- Have customers participate in Sprint Review to reinforce customer value perspective
- Have PO spend more time with Customer and/or get training on writing better user stories

Target end-state

- Each completed Story delivers a “potentially shippable” increment of value to customer
- Multiple team members can “swarm” together on priority stories
- Every Story is immediately clear and actionable
- Sprint burns down relatively smoothly
- Release Plans are relatively accurate
- Velocity is increasing roughly 10% per Sprint

Break Epics into Stories

As a frequent flyer I want to book flights customized to my preferences, so I save time



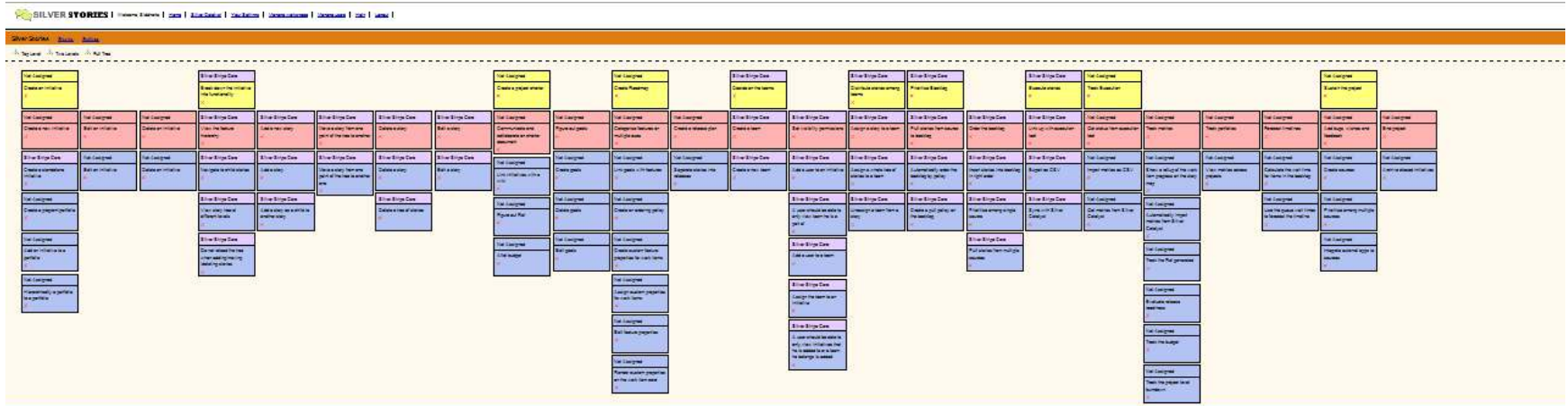
As a frequent flyer I want to book a trip using miles so that I can save money

As a frequent flyer I want to easily book a trip I take often So that I can save time

As a premium frequent flyer I want to request an upgrade So I can be more comfortable

User Story Mapping

- Epics at top, stories underneath
- Shows workflow
- Can be large features, company initiatives
- Two dimension view easier to understand than linear ordering
- Tool for identifying MVP
- Allows the team to see the big picture



Team Over-Commits to Work (or is Committed by Someone Else)

Impediments

Ready



Done

Typical symptoms

- At the end of most Sprints, there are unstarted stories or stories not meeting Definition of Done
- Team is working at a unsustainable pace to try and complete each sprint
- The number of stories “in progress” remains high throughout the sprint
- Team feels “behind schedule” or under pressure to finish more output quickly

Root causes

- Team is not completing most Sprints
 - Team over commits during Sprint Planning
 - Team guesses about how much work it can complete each sprint
 - **Team is not tracking velocity**
- Team is working at an unsustainable pace to complete each sprint
 - The team is overcommitted
 - Team following a plan that dictates what must be done by when
 - Team does not control what work is brought into the Sprint
 - **Team is not self-organizing**

What to do about it

Team not tracking Velocity

- Each story brought into the sprint should be estimated in points
- All finished points totaled at end of every Sprint
- Implement Yesterday’s Weather Pattern for Sprint Planning

Team is not self-organizing

- Align with leadership on expectations for empowered teams
- Secure buy in that reality on the ground trumps the plan
- Team estimates work and commits to how many stories to bring into the sprint

Target end-state

- Team is tracking Velocity each sprint and all team members know Velocity if asked
- Team pulls in work equal to the average actual points completed in recent sprints
- Team and PO work together to prepare for Sprint Planning
- Team decides, and is not told, how much work to pull into the Sprint Backlog

Pattern: Yesterday's Weather

How much work to pull into the Sprint



To Do	WIP	Done
3 1	5	<div><div>8</div><div>8</div><div>5</div><div>5</div><div>3</div><div>3</div><div>3</div><div>1</div></div>

$V=33$

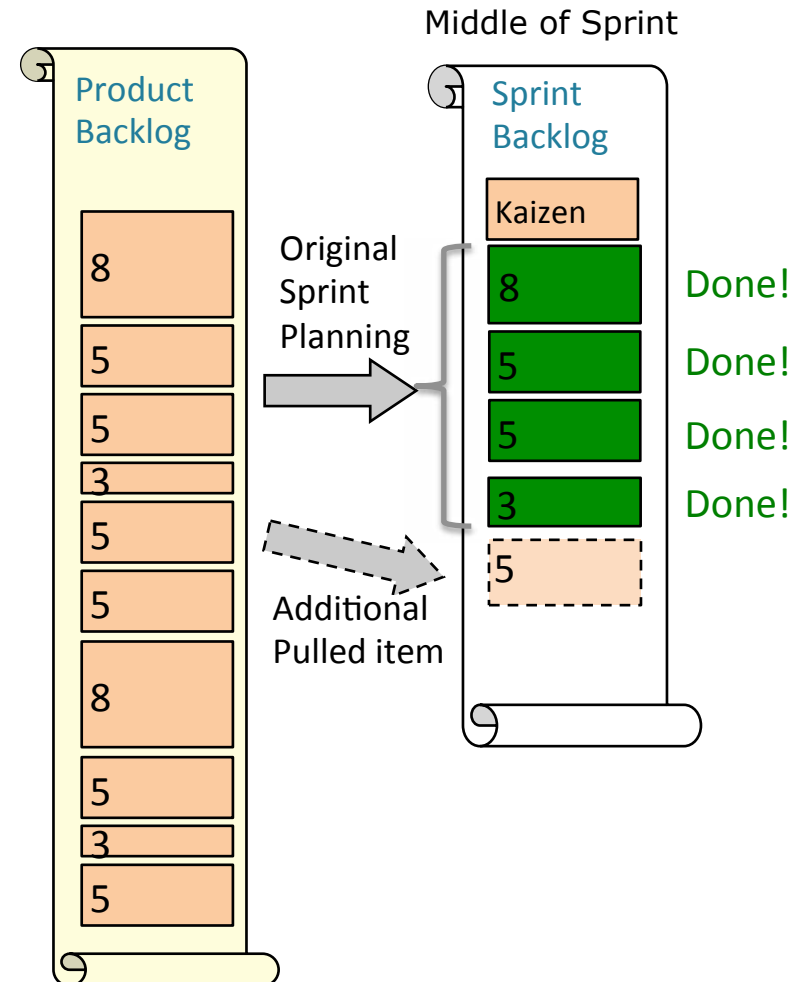
S1:33 | S2: 40 | S3: 38

Average Velocity = 37

- Start by tracking Velocity by estimating stories in points, not hours.
- At the end of the sprint, tally how many story points have met the definition of done.
- Use the average actual velocity during Sprint Planning to estimate how many points the team will likely complete in the upcoming sprint.

Pattern: Teams that Finish Early Accelerate Faster

- If team completes Sprint Backlog before end of the Sprint, they should pull the next Ready item from the top of the Product Backlog
- Velocity for the Sprint is the total points completed (including pulled stories)
- Experience shows teams that use this approach increase Velocity faster than those that try to pull too much work initially



White Paper at: <http://scruminc.com/FinishEarlyAccelerateFasterHICSS2014.pdf>

Team Working Individually Rather than Together

Impediments

Ready



Done

Typical symptoms

- Team thinks of backlog as a shared “to do” list where each PBI is done by only one person: “those are *my* stories”
- Team comprised of Subject Matter Experts
- Bottlenecks created around a single Team member
- One person or group typically working long hours to keep up with demand on their time

Root causes

- High level of Work in Progress (WIP)
 - Each team member pulls a different story
 - Stories requires skill only one Team member possesses
 - Lower priority stories started before higher priority ones completed
 - Next available Team member can’t pull next high priority story
 - High priority story depends on scarce skill
 - **Need for cross-training on skill**
 - Team often relies on one hero to “save the day”
 - This person is only one who can do a task
 - **Team works as a group of individuals**

What to do about it

Pair on Stories

- Encourage collaboration on stories to increase the quality of the end product
- Write stories that provide opportunities to pair
- “Divide and conquer” to get Done on priority stories quicker

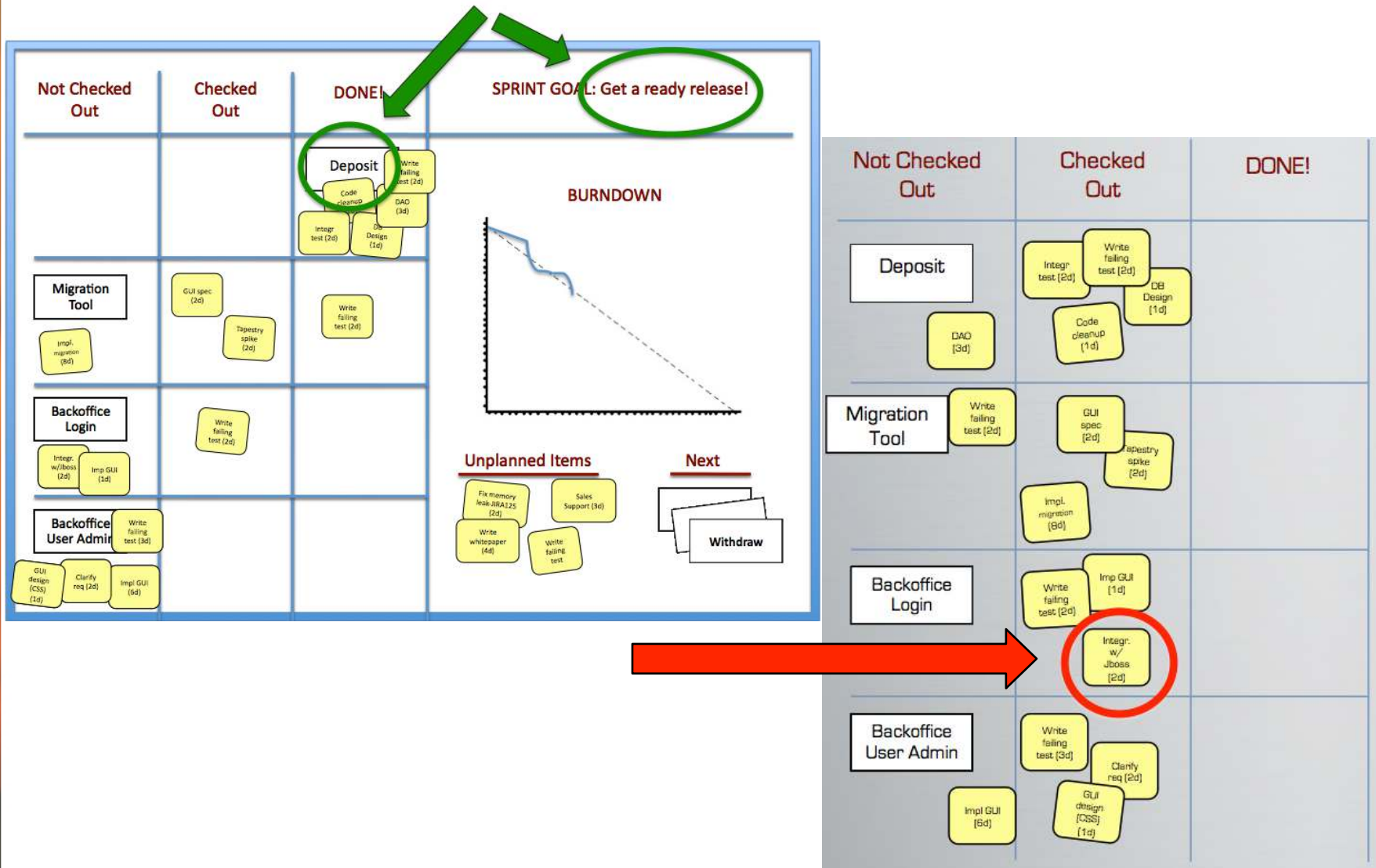
Cross-Train to grow Team’s skillset

- Flag scarce skills as a Team impediment
- SME works with one or two Team members to help them learn the unique skill
- Lightweight checklists or notation stored in a Team Wiki for reference for common tasks

Target end-state

- A least two Team members can finish each story and ideally anyone can work on any story
- Work in Progress is low as the Team works together on top priority stories
- Work flows easily from one to member to another
- Team members can enjoy vacation without being needed to deliver work!

Value to "Swarming" on the Backlog



Source: Revised after Henrik Kniberg

Context Switching Kills Productivity

Weinberg's Table of Project Switching Waste

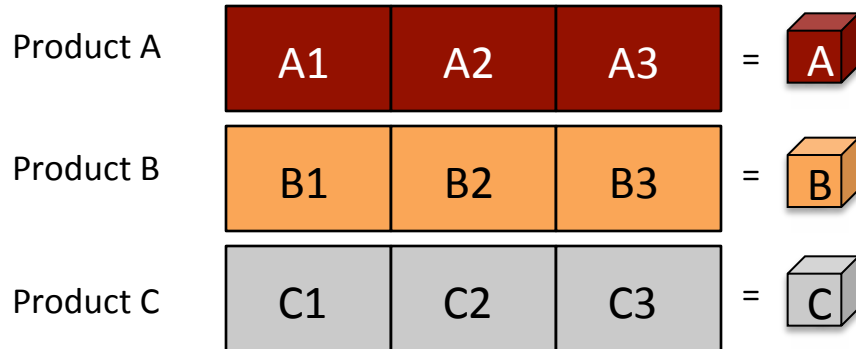
Table 2-1: Waste Caused by Project Switching

Number of Simultaneous Projects	Percent of Working Time Available per Project	Loss to Context Switching
1	100%	0%
2	40%	20%
3	20%	40%
4	10%	60%
5	5%	75%

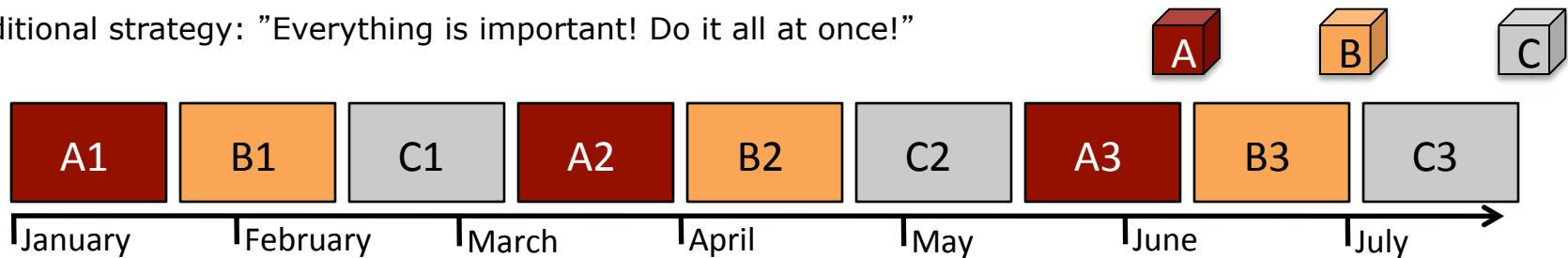
Weinberg, Gerald M. (1992) Quality Software Management: Systems Thinking. Dorset House, p. 284.

Pattern: Swarming

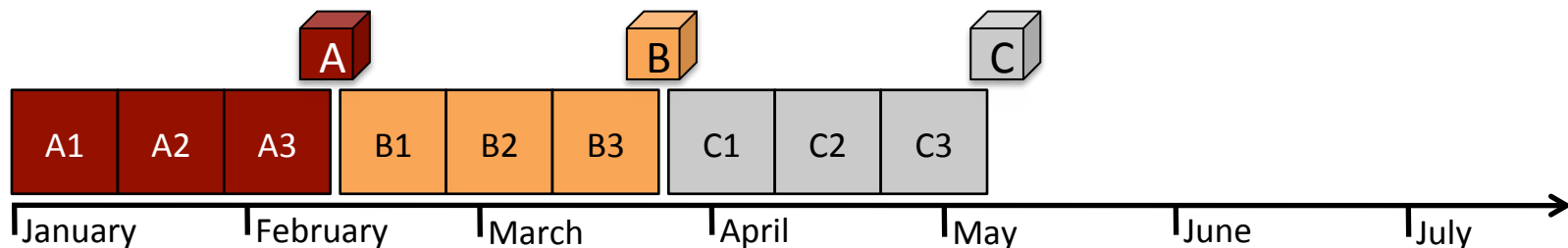
Prioritizing Between Projects



Traditional strategy: "Everything is important! Do it all at once!"



Agile strategy: "Prioritize & focus!"



Adapted from Henrik Kniberg

Found Work Interrupts Sprint Regularly

Impediments

Ready



Done

Typical symptoms

- Team frequently (>20%) fails to complete planned work by end of Sprint
- Team discovers significant unplanned work or receives frequent “surprise” requests from stakeholders that must be addressed right away
- Team feels like priorities are constantly shifting
- Planned stories don’t move to Done
- Burndown chart is flat

Root causes

- Significant amounts of “found work” enters sprint
 - Team not anticipating what is needed to complete work
 - Team is new, or working in unfamiliar area
 - Team hasn’t given room in Sprint for learning
 - **Build in “buffer” for found work**
- Frequent surprise requests from stakeholders
 - Stakeholders asking Team directly for work
 - No formal process for handling “urgent” requests – informal requests add up
 - **Need process for managing, prioritizing and limiting mid-sprint external requests**

What to do about it

Found works interrupts Sprint regularly

- Implement the Interrupt Pattern and include Sprint buffer in categories where found work is expected
- Use Retrospective to identify ways to anticipate found work better

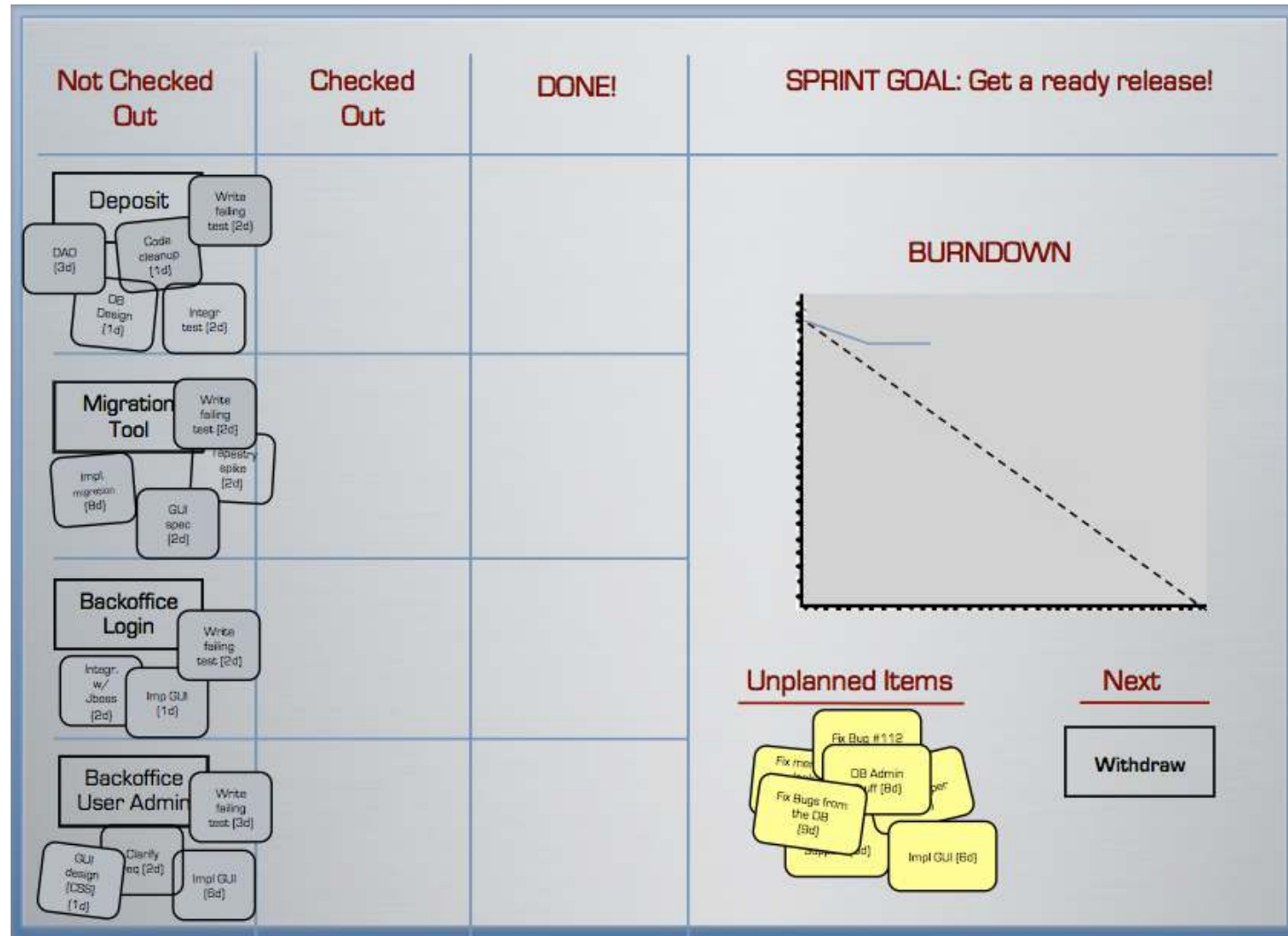
External stakeholder requests displace planned work

- Confront leadership with the effect of interruptions
- Implement the Interrupt Pattern, include limited buffer for surprise requests, and put PO in path to defend team

Target end-state

- Team anticipates some level of unplanned work, and allows for this in Sprint Backlog
- Unplanned work is limited to allow planned work to proceed to completion
- Team finishes all planned work early, and is able to pull additional stories from Product Backlog
- Velocity increases ~10% each Sprint as planning and execution improve

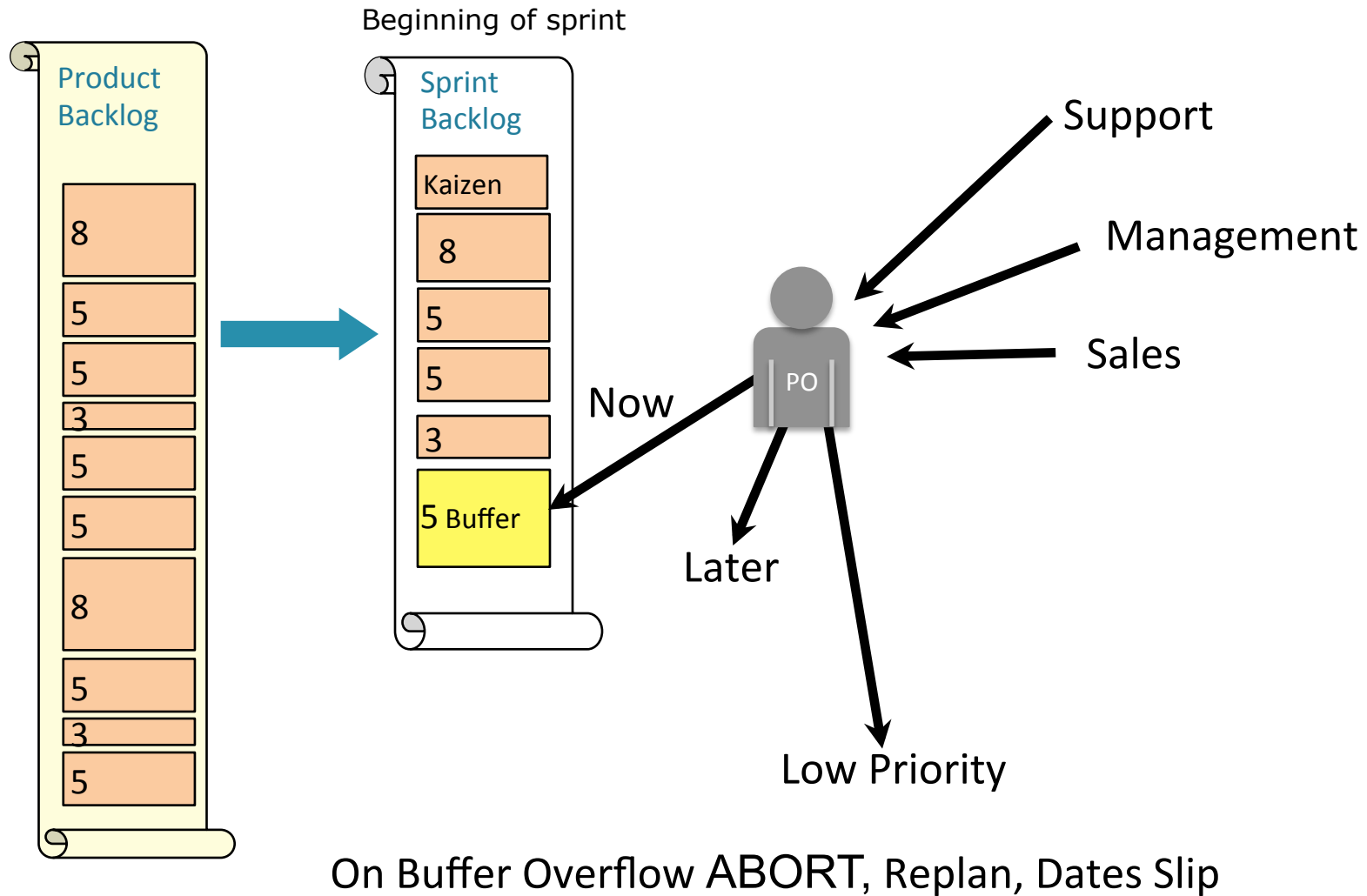
If Your Backlog Looks Like This, You Have a Problem with Interrupts!



Source: Revised after Henrik Kniberg

Pattern: The Interrupt Pattern

Dealing with the unexpected



Conclusion

- We have reviewed the seven pitfalls we encounter most frequently in the field
- If any of the symptoms above sound familiar, you may be experiencing one of these challenges now
 - Conduct an A3 to flesh out root causes and align organization around a plan of action
 - We are always happy to help
- Addressing READY and DONE well should lead to at least a doubling of Velocity
- A future webinar will address pitfalls with removing impediments and securing leadership support

Stay Connected

Our Website

- check in for announcements, new content and services, book releases, and more!
- www.scruminc.com

ScrumLab

- join the conversations on our forums with the scrum community and your class.
- coming soon: articles, videos, papers on all things scrum
- scrumlab.scruminc.com

Blog

- scrum.jeffsutherland.com

Webinars

- advance your learning with our interactive webinars. visit the scrumlab store to view upcoming topics.

Twitter, Facebook, and G+

- @jeffsutherland, scrum and scrum inc.