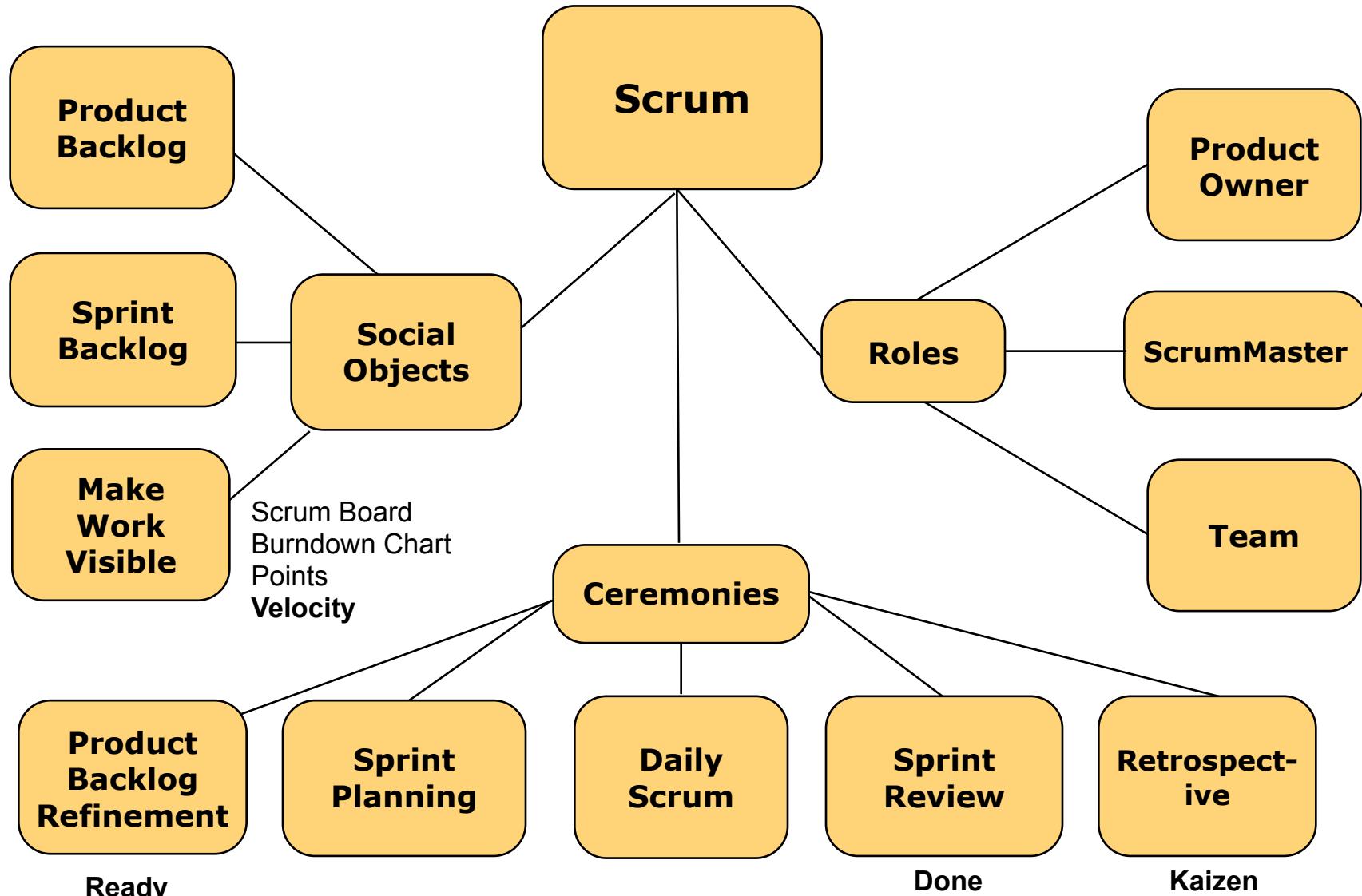


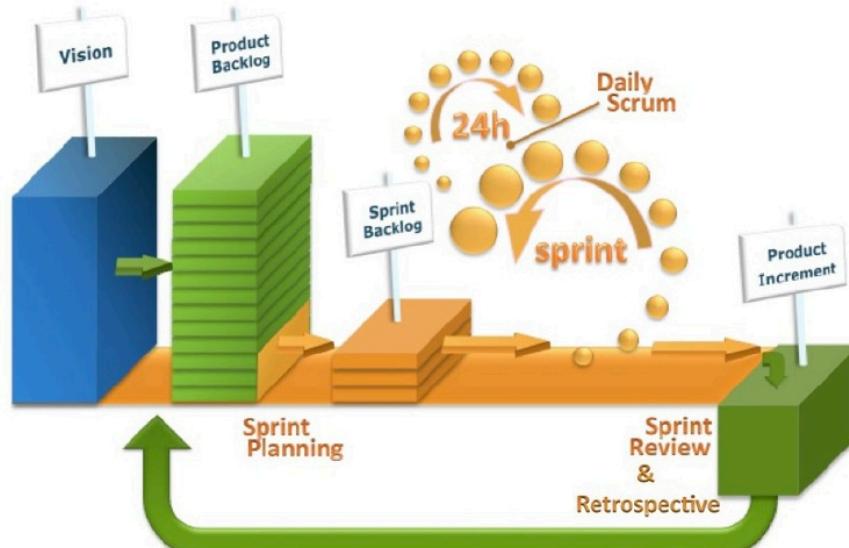
What is Scrum?

Scrum Framework

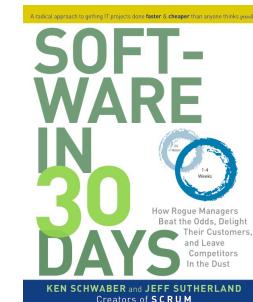


Scrum Framework in 30 seconds

- A product owner creates a prioritized wish list called a product backlog.
- During sprint planning, the team pulls a small chunk from the top of that wishlist, a sprint backlog, and decides how to implement those pieces.
- The team has a certain amount of time, a sprint, to complete its work - usually two to four weeks - but meets each day to assess its progress (daily scrum).
- Along the way, the ScrumMaster keeps the team focused on its goal.
- At the end of the sprint, the work should be potentially shippable, as in ready to hand to a customer, put on a store shelf, or show to a stakeholder.
- The sprint ends with a sprint review and retrospective.
- As the next sprint begins, the team chooses another chunk of the product backlog and begins working again.



Source: Gunther Verheyen / Capgemini 2011



Scrum has Three Roles

- Product Owner:
 - Define and prioritize the features of the Product Backlog
 - Decide on release date and content
 - Responsible for the profitability of the product (ROI)
- ScrumMaster
 - Facilitates the Scrum process and Team self-organization
 - Removes obstacles and shields the team from interference
 - Responsible for improving performance of the team
- Team
 - cross-functional (incl. testing)
 - self-organizing/-managing group of individuals, has autonomy regarding how to achieve its commitments
 - typically 3-9 people

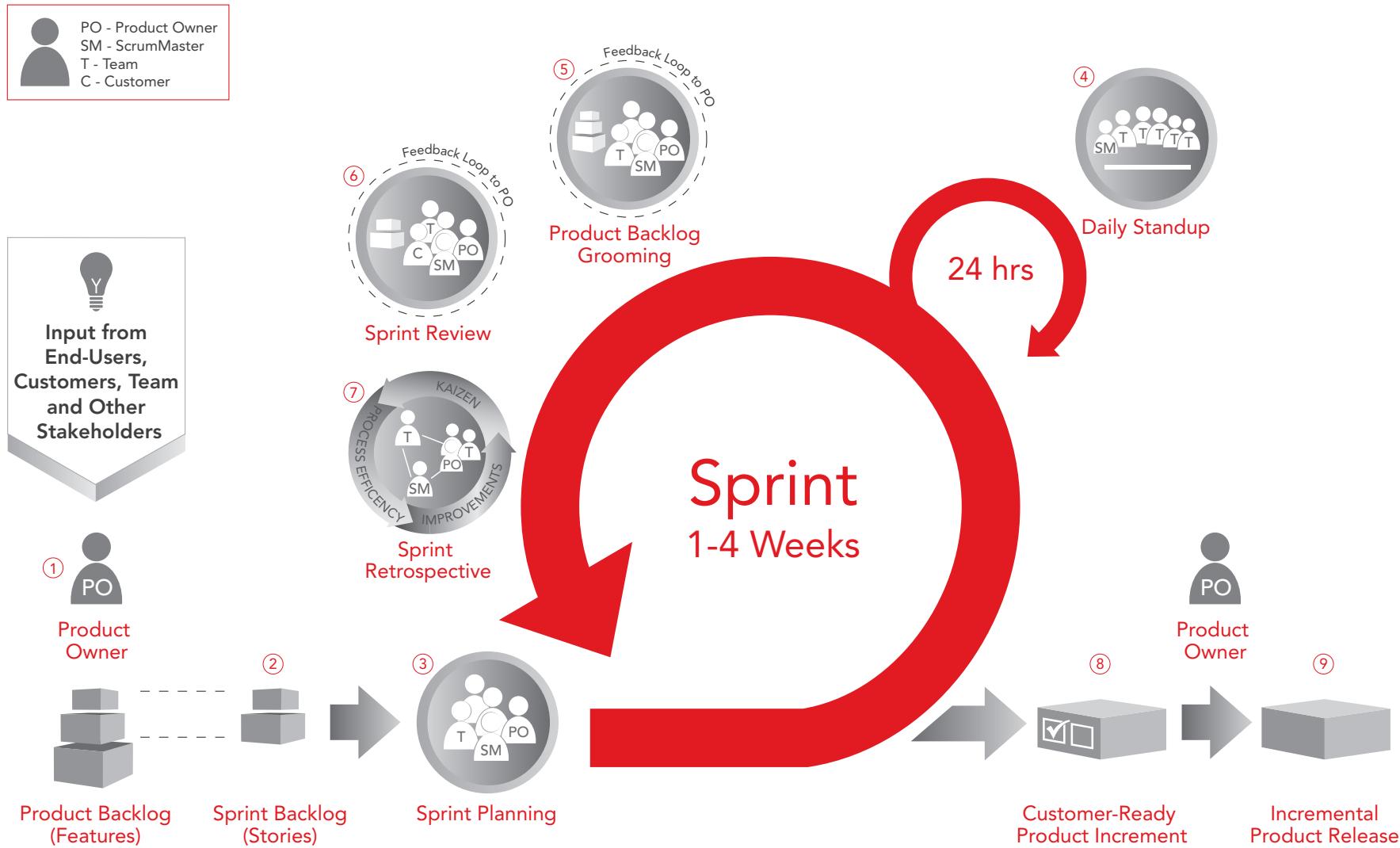
Scrum has Four Meetings

- Sprint Planning
 - Product Backlog must be READY before this meeting
- Daily Scrum
 - Self-organize to improve performance
- Sprint Review
 - Decide what is DONE. That determines velocity.
- Retrospective
 - Identify the top process improvement and put in the backlog for the next sprint

Scrum Makes Work Visible

- Product Backlog
- Sprint Backlog
- Scrum Board
- Burndown Chart
 - Show work remaining
 - Velocity

How Scrum Works



Teams are:

- Cross-functional
- Self-organizing
- Self-managing

Stable (Brooks)Law

Simple Rules Create Self-Organization



- Simple, clear purpose and principles give rise to complex, intelligent behavior. Complex rules and regulations give rise to simple, stupid behavior. --
Dee Hock, VISA

Agile Manifesto

www.agilemanifesto.org

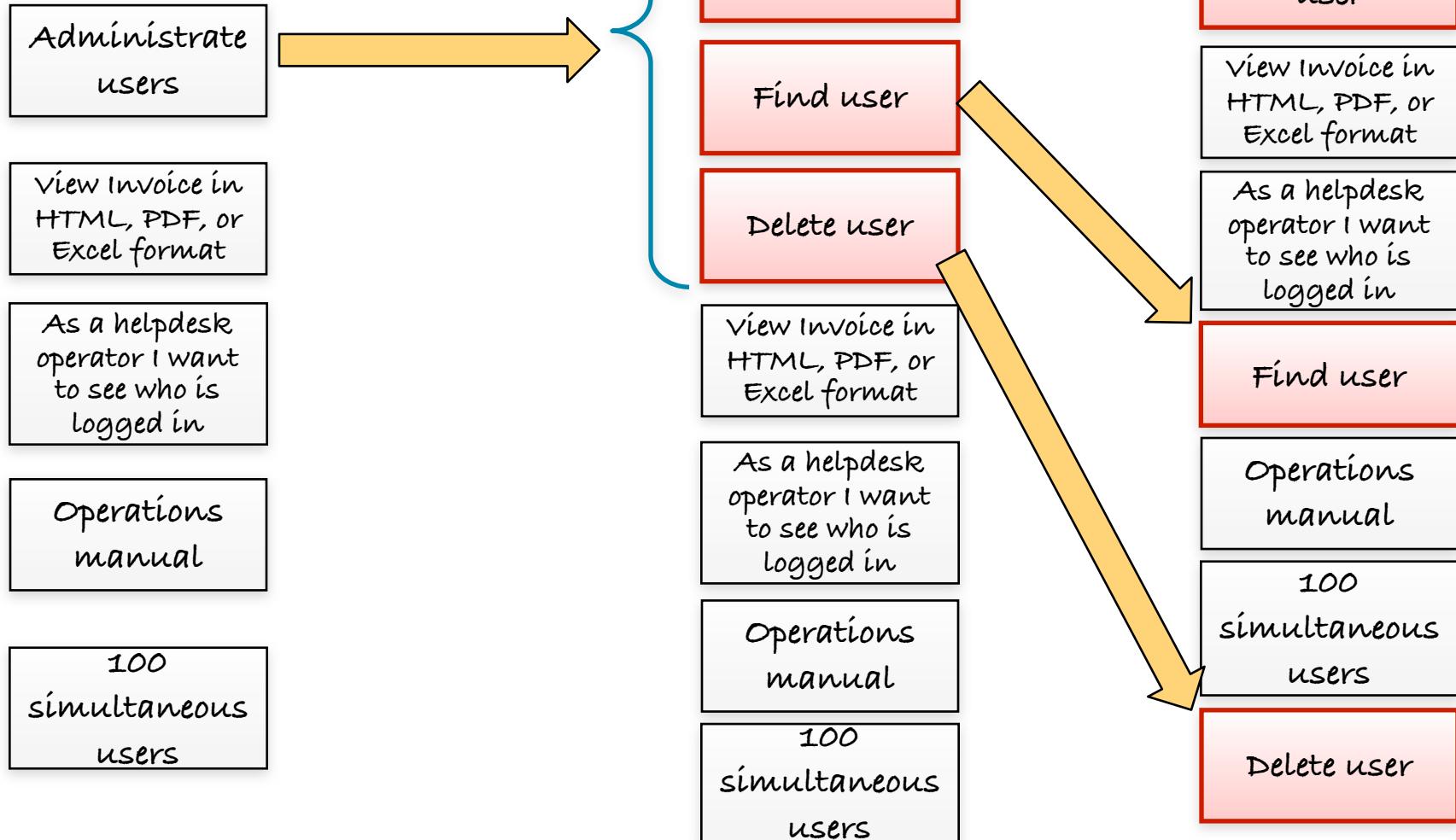
We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

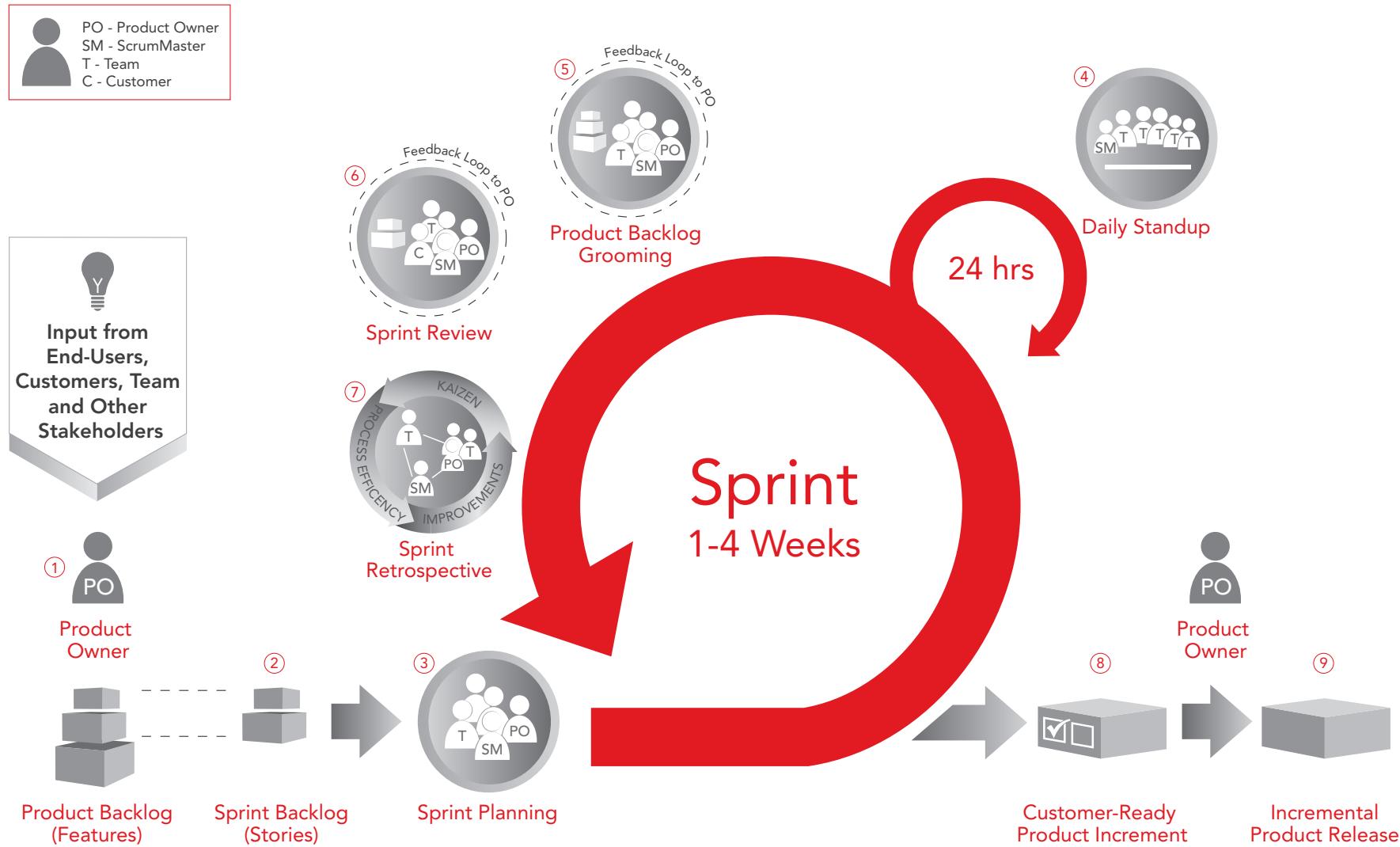
Product Backlog and User Stories

Product Backlog



Source: Henrik Kniberg

How Scrum Works



User Story

- A UserStory is a story, told by the user, specifying how the system is supposed to work, written on a card, and of a complexity permitting estimation of how long it will take to implement. The UserStory promises as much subsequent conversation as necessary to fill in the details of what is wanted. The cards themselves are used as tokens in the planning process after assessment of business value and [possibly] risk. The customer prioritizes the stories and schedules them for implementation.

-- RonJeffries

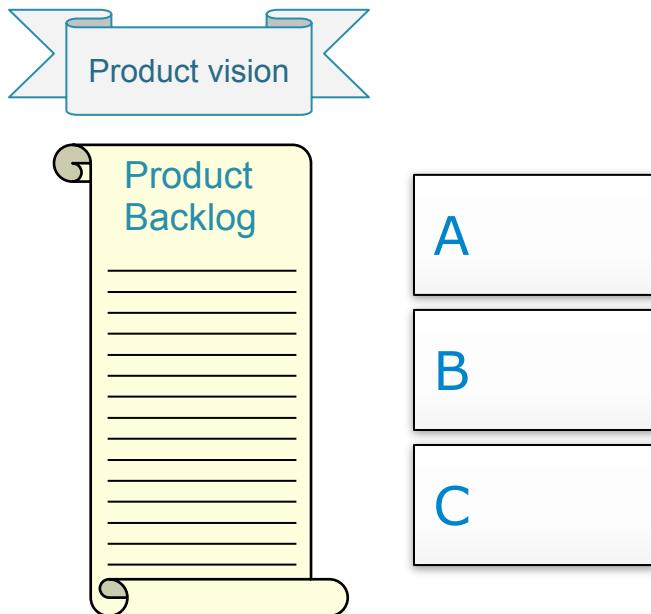


User Story Templates

- As a <role> I would like to be able to <action> to achieve <business value>
- As a <user role> I can <story> so that <benefit>
- As a <person name> can < story > so that <benefit>
- As a <user> I want to <goal> so that <value to attain>

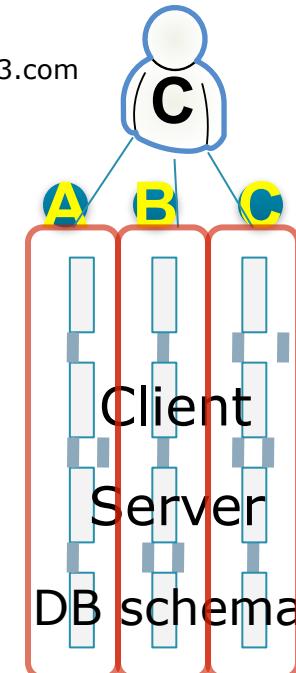
The “so that” line is generally considered optional, but used as a default

User Story - Guidelines



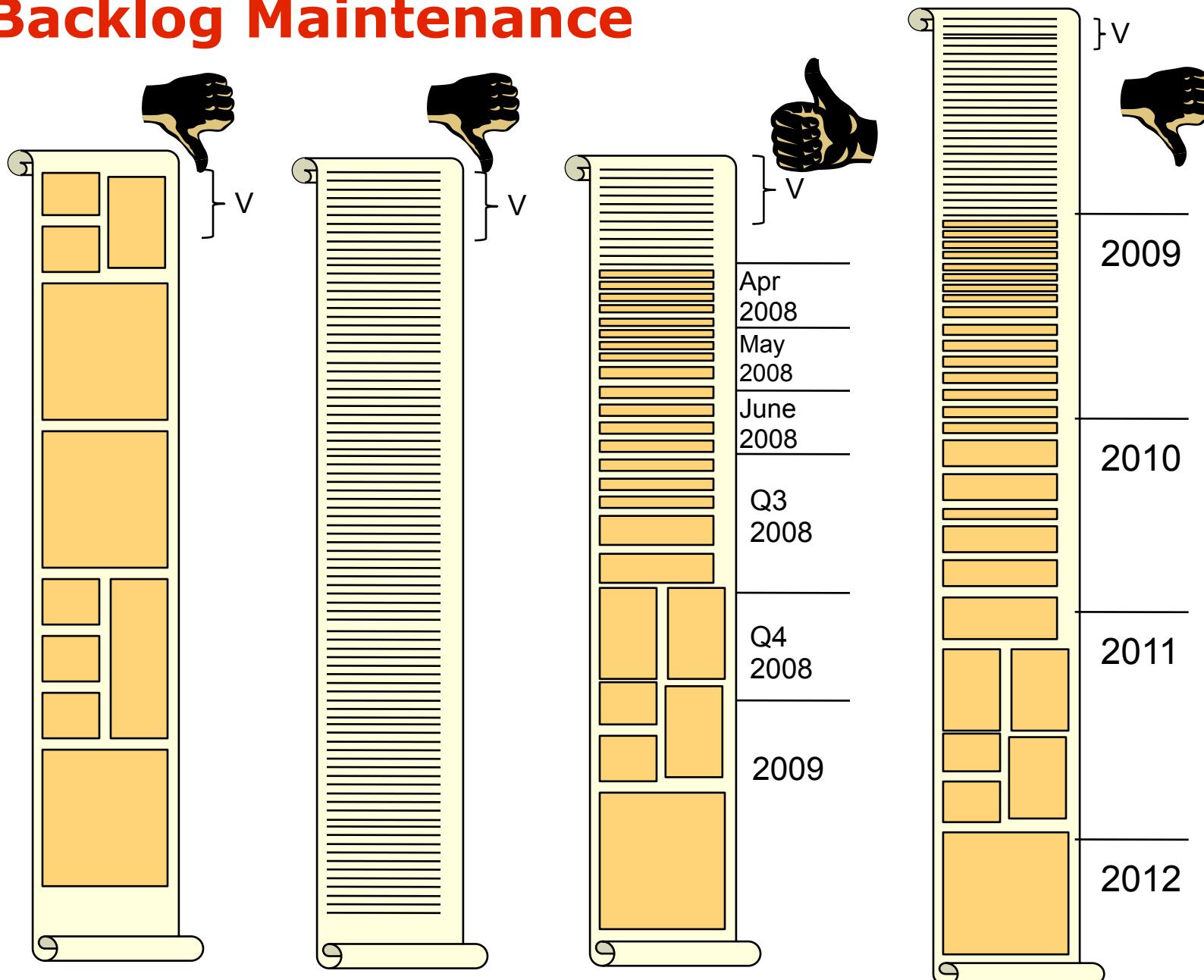
Immediately actionable
Negotiable
Valuable
Estimable
Sized to fit
Testable

Modified from Bill Wake – www.xp123.com



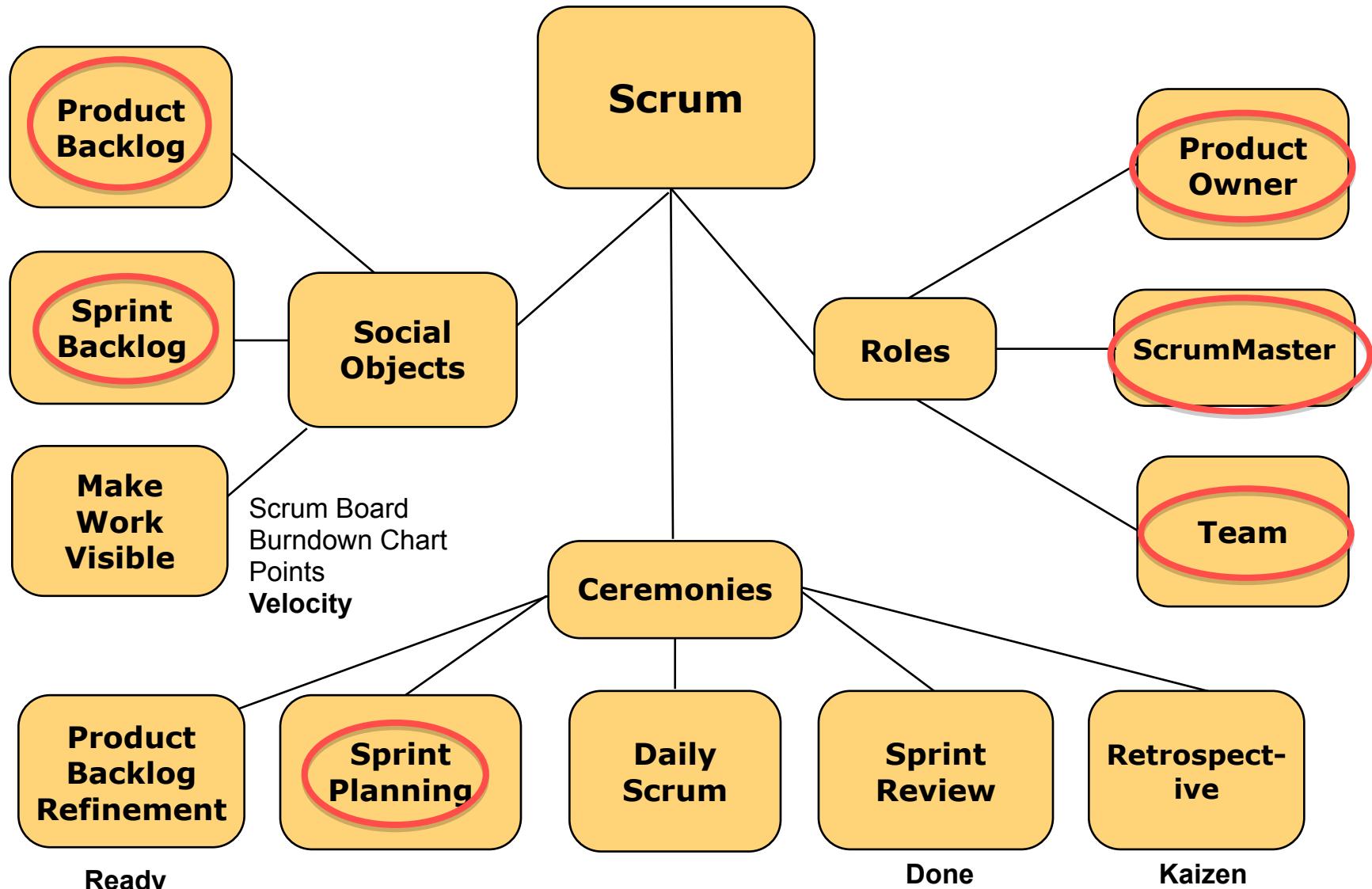
Refining the Product Backlog

Backlog Maintenance

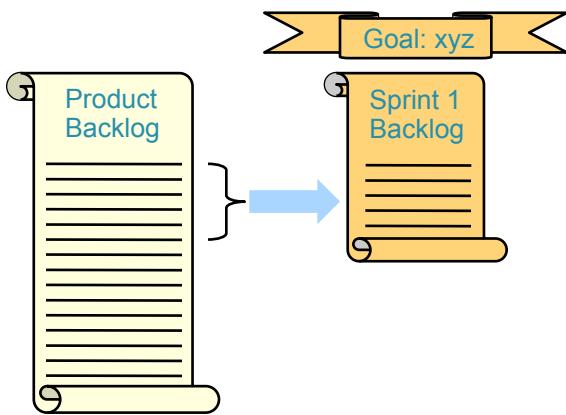


Sprint Planning

Scrum Framework



Sprint Planning Meeting



Jackass team, sprint 15

Sprint goal

- Beta-ready release!

Sprint backlog

- Deposit (5)
 - Migration tool (13)
 - Backoffice login (3)
 - Backoffice user admin (5)
- (Estimated velocity = 26)

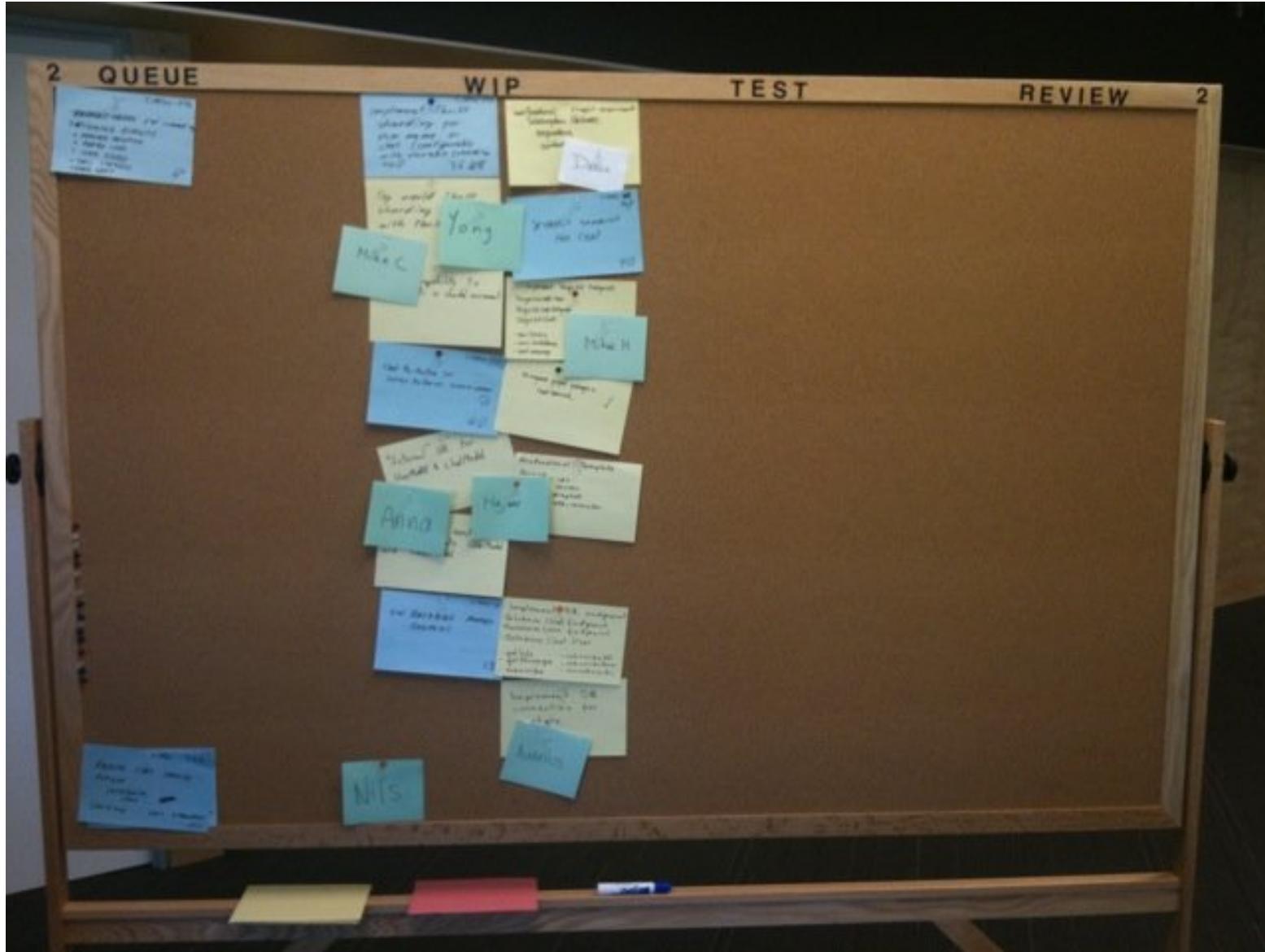
Schedule

- Sprint period: 2006-11-06 to 2006-11-24
- Sprint demo: 2006-11-24, 13:00, in the cafeteria
- Daily scrum: 9:30 – 9:45, in conference room Jimbo

Team

- Jim
- Erica (scrum master)
- Tom (75%)
- Niklas
- Eva
- John

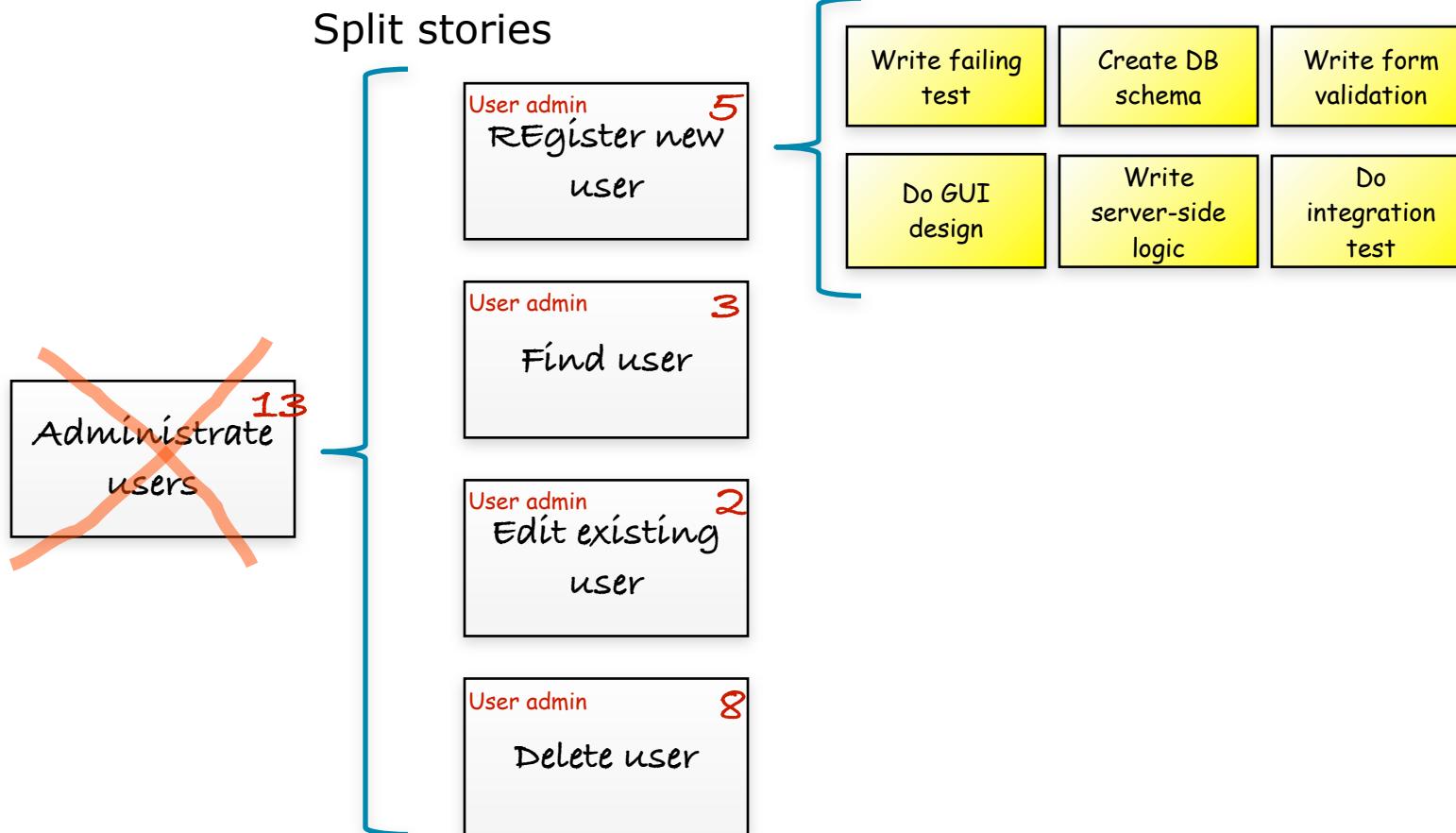
Swimlane Scrum



Source: Jim Coplien

Splitting Stories and Breaking Out Tasks

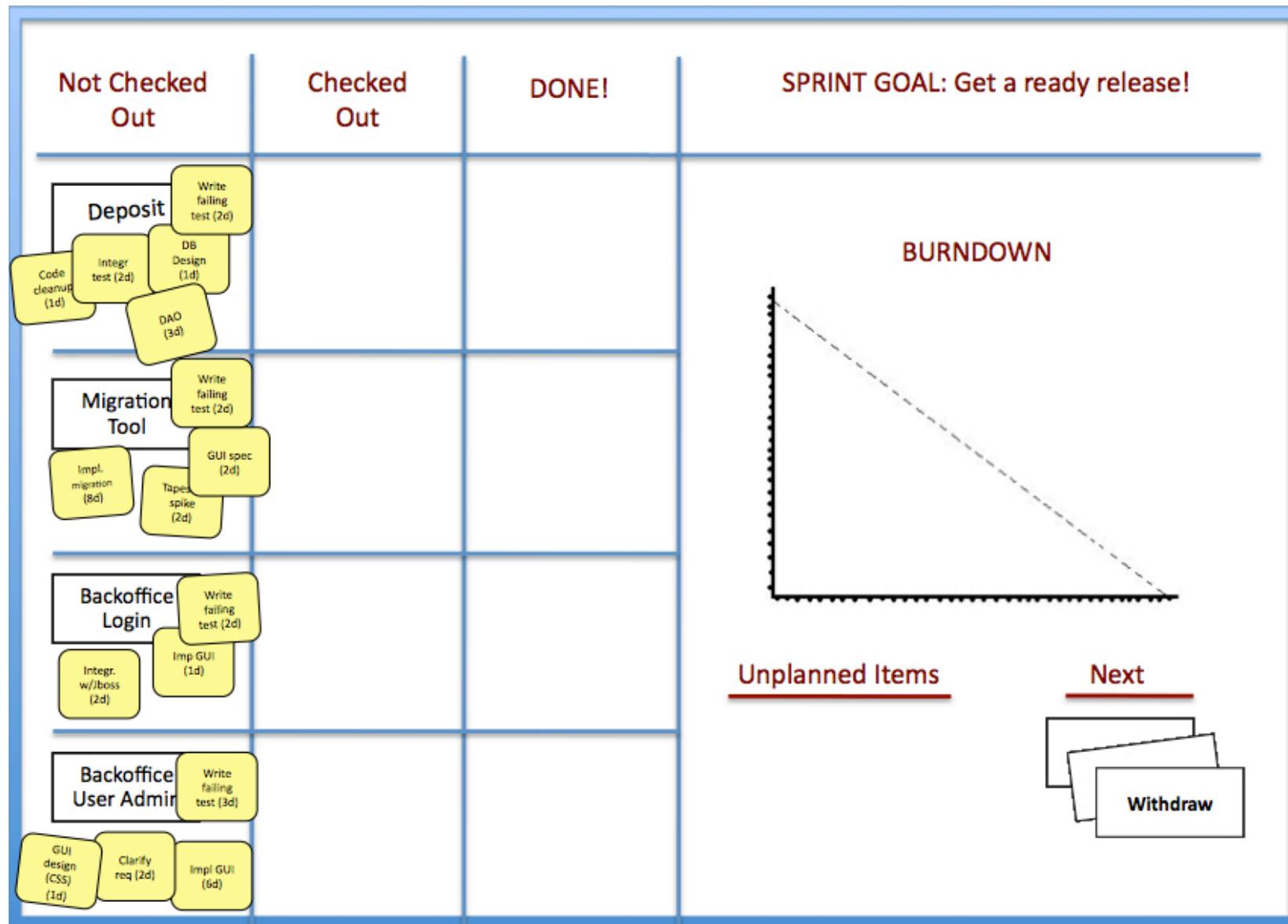
Break into tasks
(normally during sprint planning meeting)



Source: Henrik Kniberg

© 1993-2012 Jeff Sutherland

Sprint backlog – day 0



Source: Revised after Henrik Kniberg

© 1993-2012 Jeff Sutherland

Some Definitions of Done

Default Definition of Done

- Releasable

Default Definition of Done

- Unit/Integration tested
- Ready for acceptance test
- Deployed on demo server

Default Definition of Done

- Acceptance tested
- Release notes written
- Releasable
- No increased technical debt

= I haven't messed up
the codebase

What's else must be done before shipping the code?

- For example "customer acceptance test + user documentation"

Why not? Who does it? When? What happens if a problem turns up?

Burn up this work in release burndown!

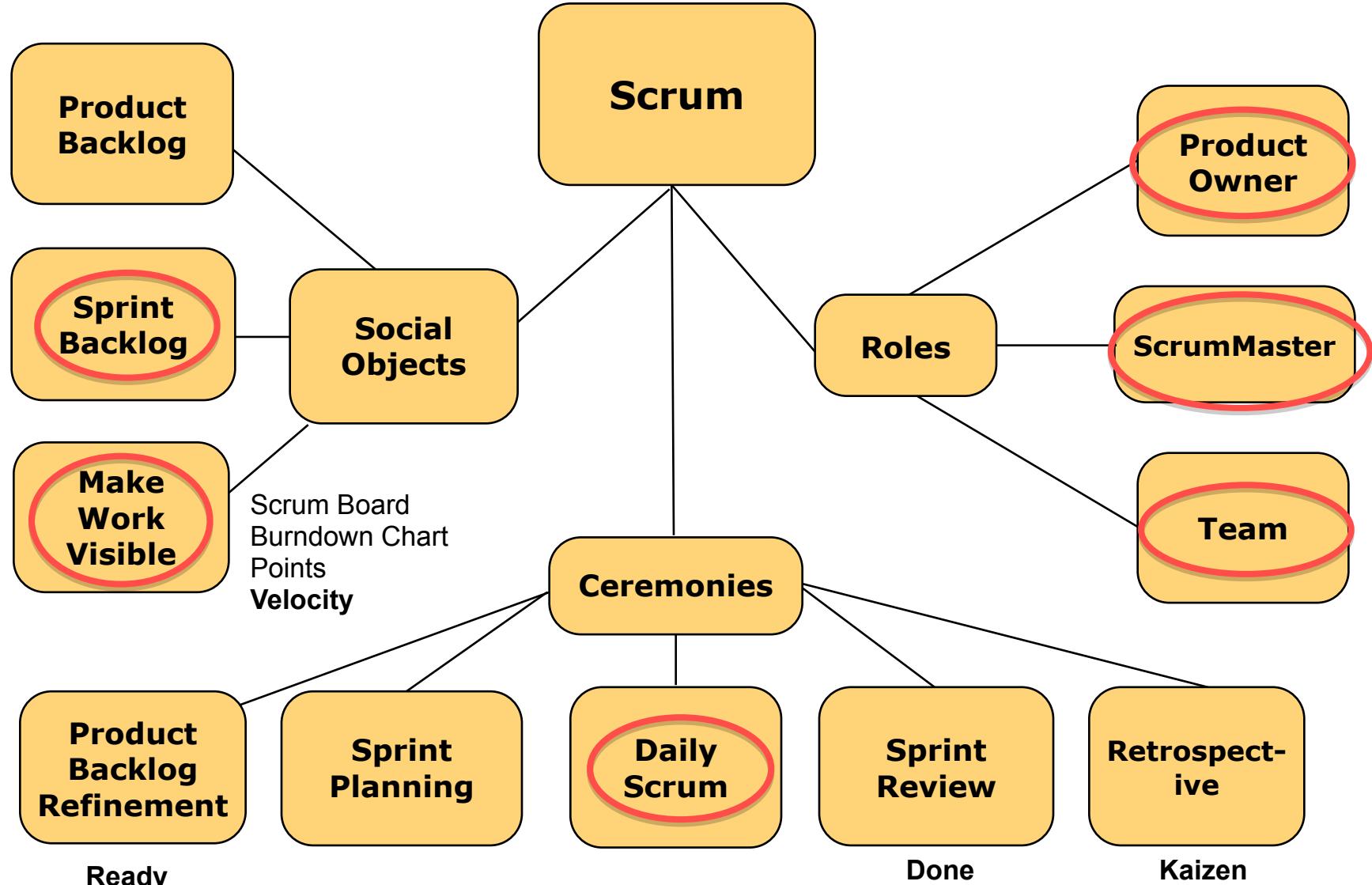
The Sprint Commitment

- Team's commitment to the Product Owner:
- "We promise that ..."
 - We believe we can reach the sprint goal.
 - We will do everything in our power to reach the goal and will inform you immediately if we have problems.
 - Code will be potentially shippable at the end of the sprint.
 - If we fall behind schedule we will remove the lowest priority stories first.
 - If we get ahead of schedule we will add stories from the product backlog in priority order.
 - We will display our progress and status on a daily basis.
 - Every story we do is complete.
- Caveat
 - Estimates are estimates. We will be early some times and late other times. We will document this normal variation with our sprint velocity.

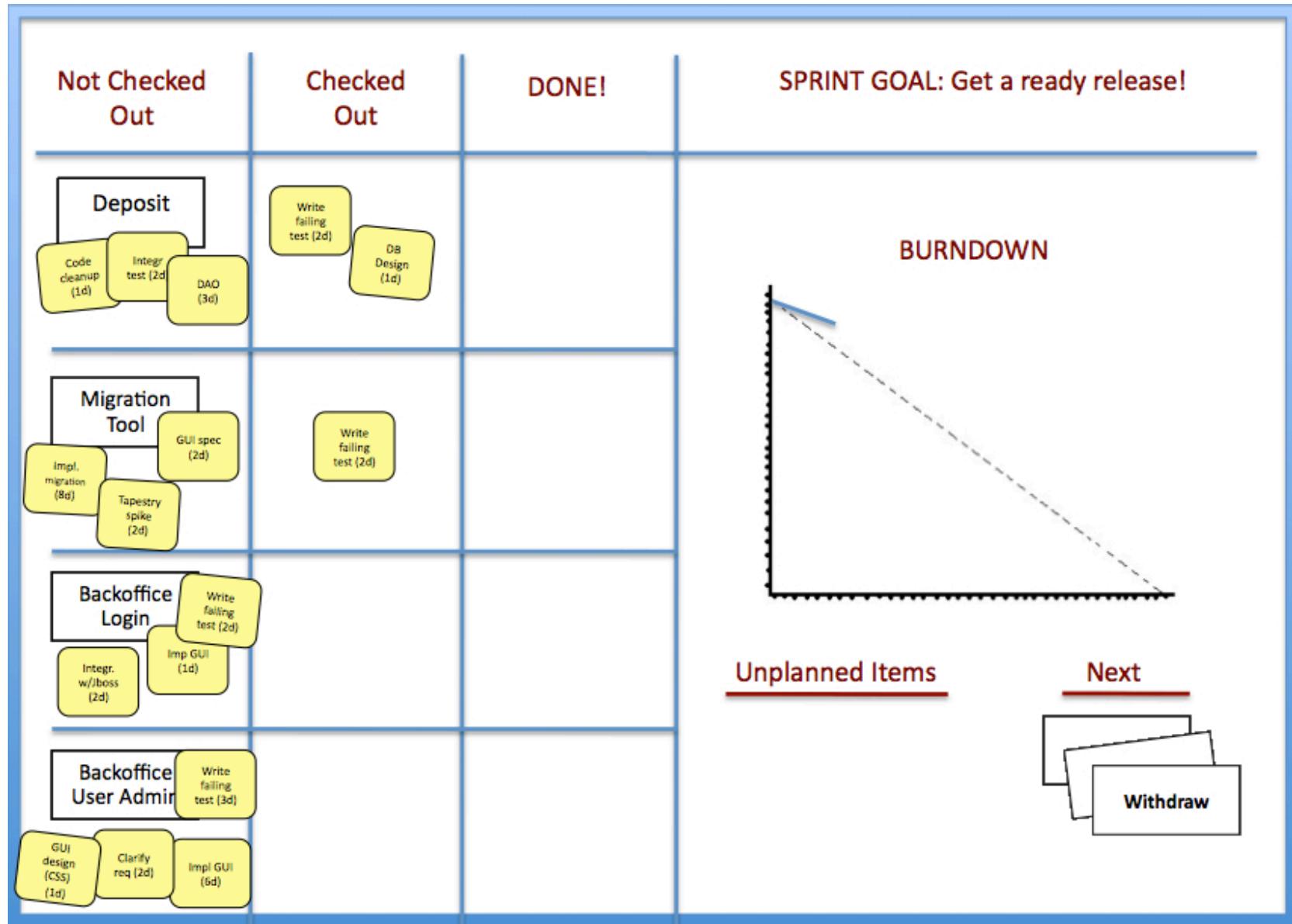
Agile Estimating Strategy

- Don't estimate time
 - Estimate relative size of stories
 - Measure velocity per sprint
 - Derive release plan
- Estimates are done by the people who are going to do the work
 - Not by the people who want the work done
- Estimate continuously during the project, not all up front
- Prefer verbal communication over detailed, written specifications

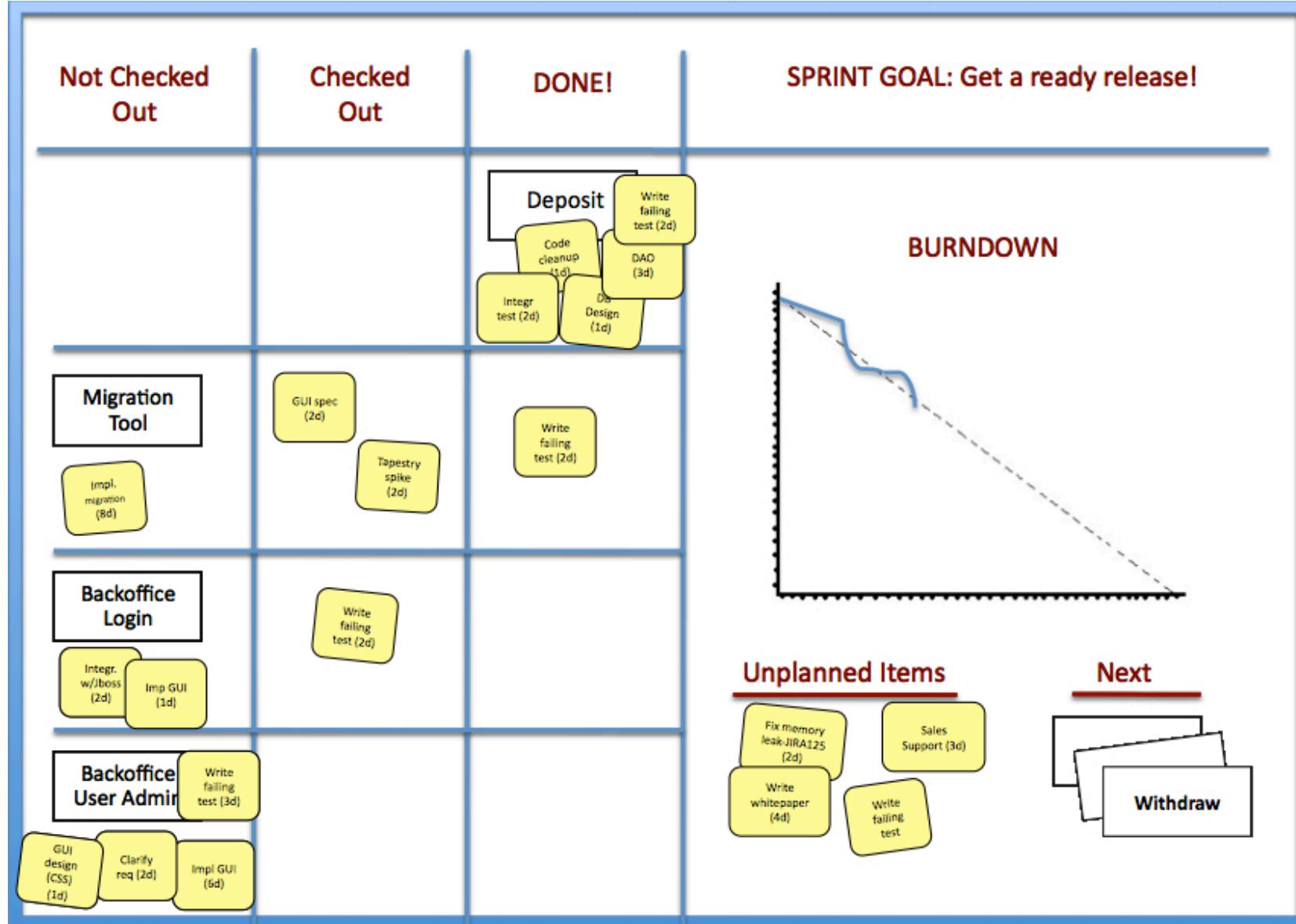
Running Sprint Scrum Framework



Sprint backlog – after 1st meeting



Sprint backlog – day X



Source: Revised after Henrik Kniberg

© 1993-2012 Jeff Sutherland

Burndown Strategies

- Burndown number of stories completed
 - stories must be small and of similar size
- Burndown stories only in points
 - stories must be small
- Burndown tasks in points
 - use planning poker to estimate tasks in points
- Burndown tasks in hours
 - Scrum Foundation and ScrumInc. no longer view this as best practice

Scrum Starter Kit

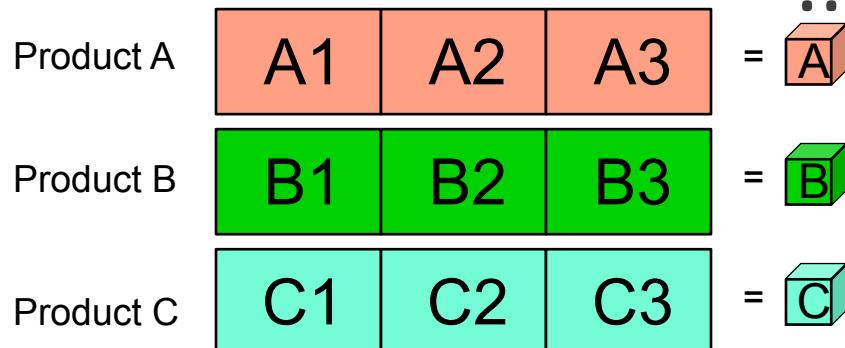
Patterns that will prevent common problems

published at scrumlop.org

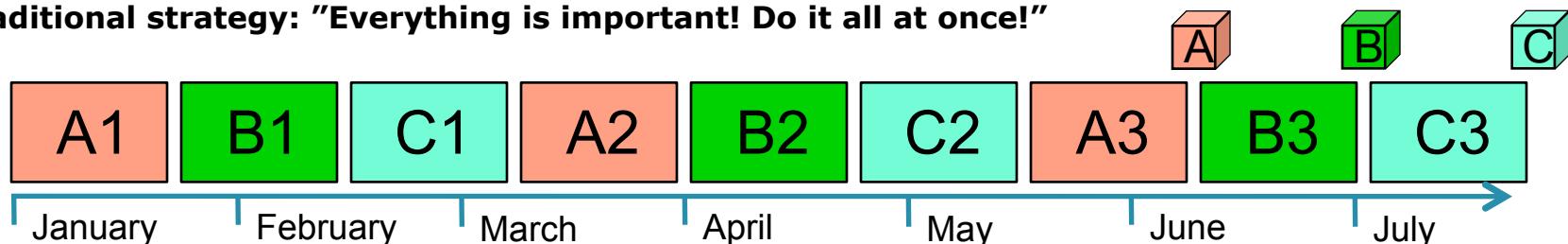
1. How do you get started? (**Stable Teams**)
2. How do you successfully pull backlog into a sprint?
(**Yesterday's Weather**)
3. How do get defect free at the end of the sprint? (**Daily Clean Code**)
4. How do you get stuff done? (**First Things First**)
5. How do you deal with interruptions during the sprint?
(**Illigitimus non Interruptus**)
6. How do you deal with emergencies? (**Scrum Emergency Procedure**)?
7. How do you ensure you continuously improve?
(**Scrumming the Scrum**) (**Happiness metric**)
8. How do you get hyperproductive? (**Teams that Finish Early Accelerate Faster**)

Prioritizing Between Projects

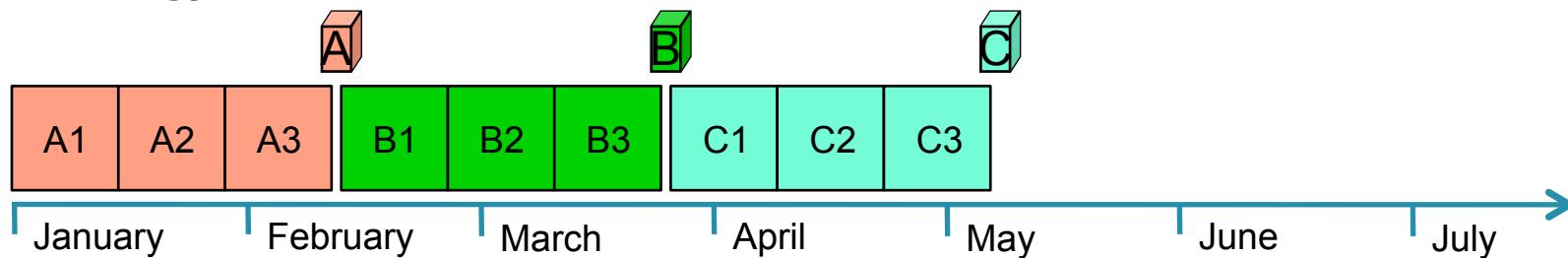
... and stories / tasks



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

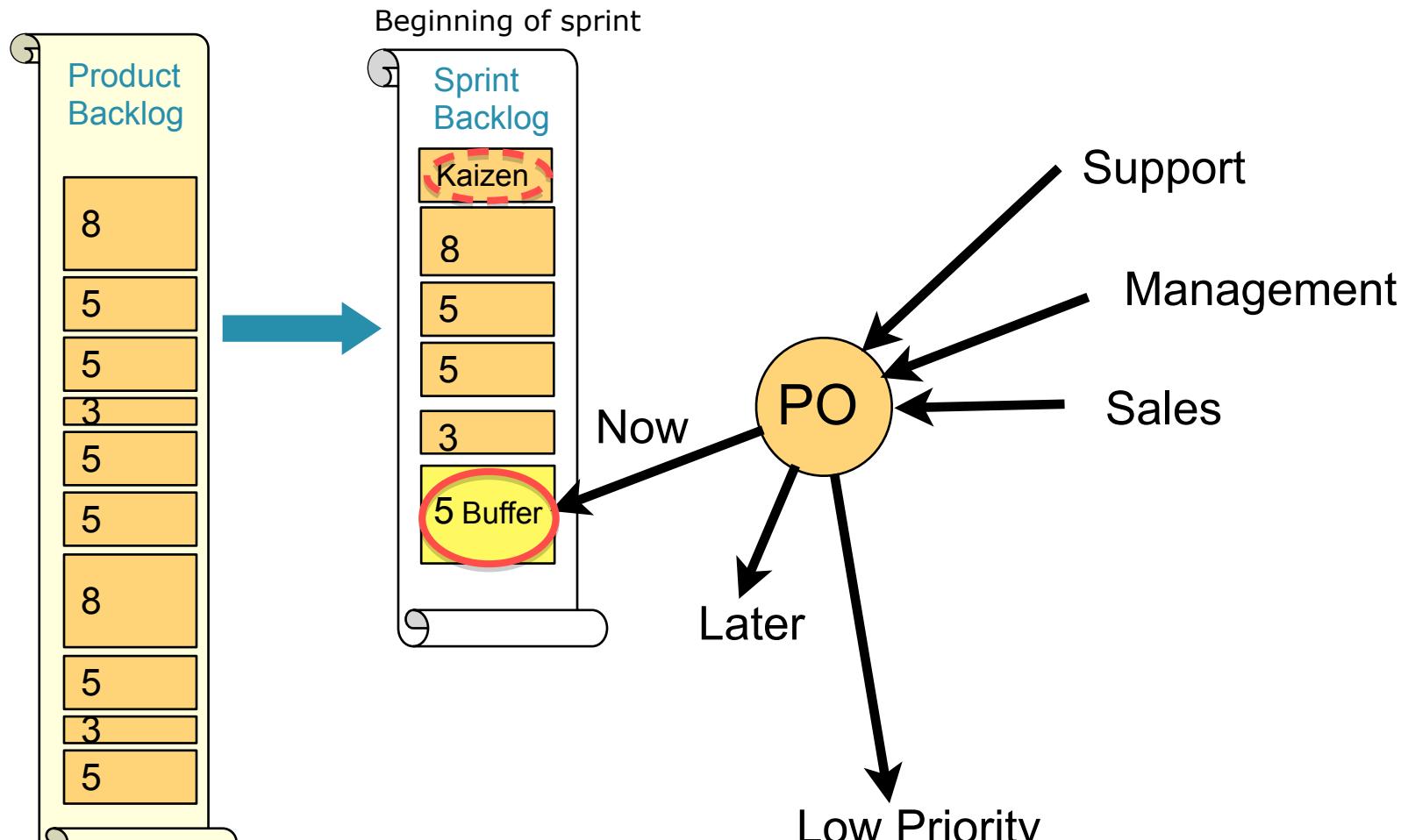


Agile strategy: "Prioritize & focus!"



Adapted from Henrik Kniberg

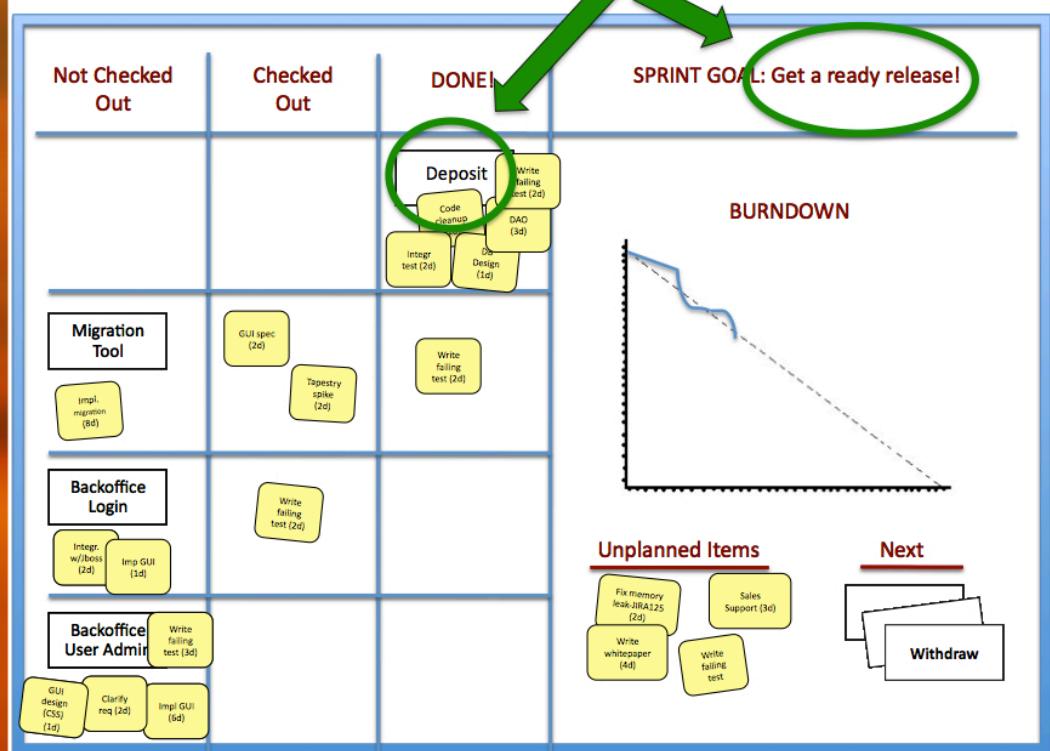
Pattern: Illigitimus Non Interruptus



On Buffer Overflow **ABORT**, Replan, Dates Slip

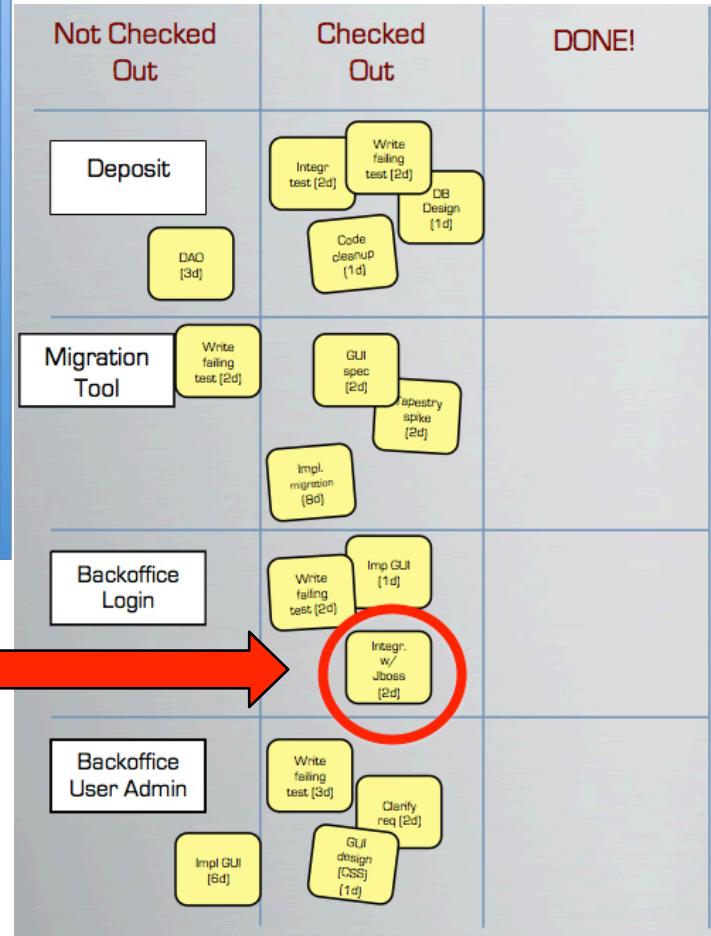
Scrum Board Warning Signs

Care about the whole product



Not just your little task

I'm more efficient if I just do my tasks

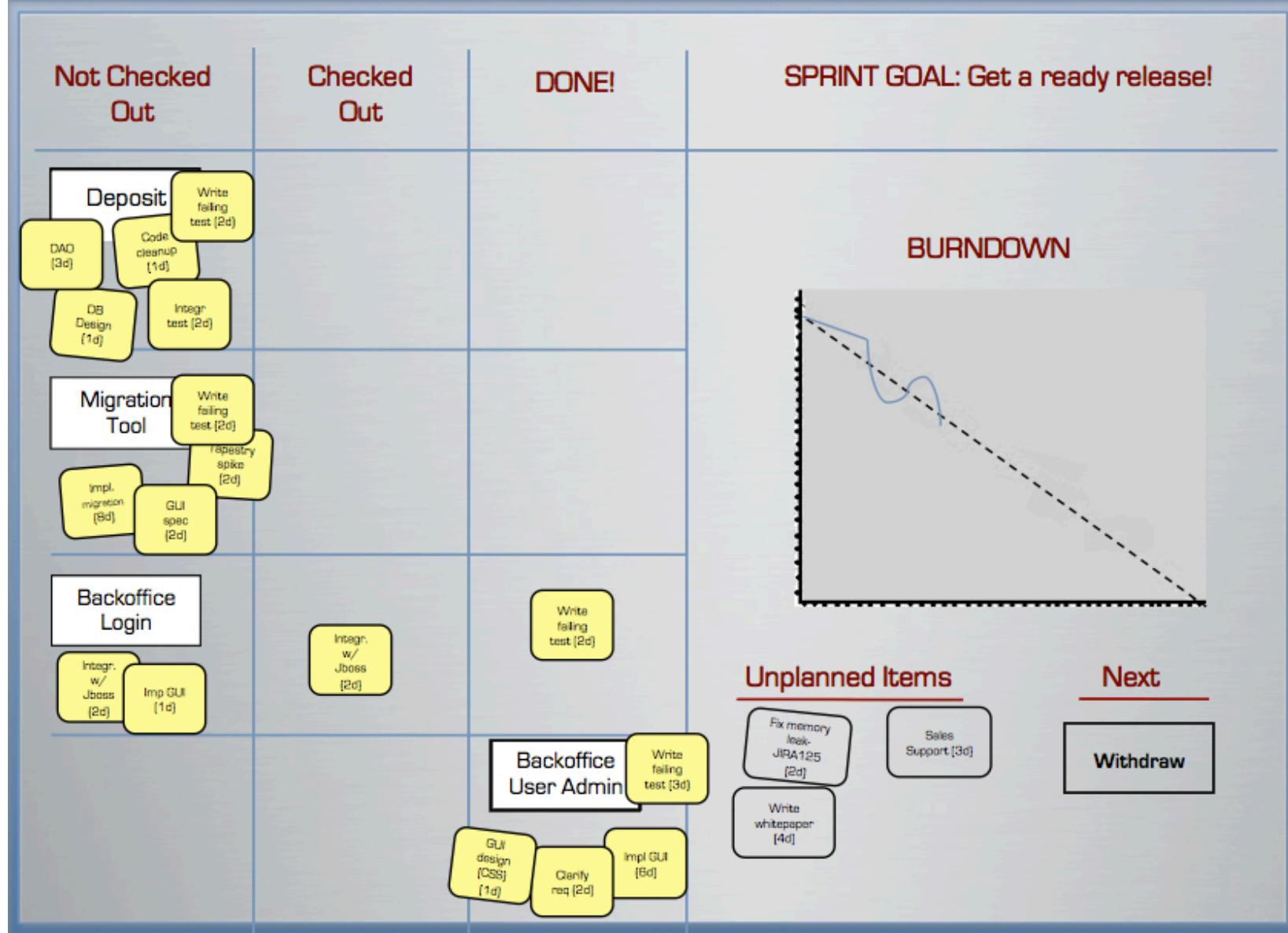


This product rocks!



Boy are we effective as a team!

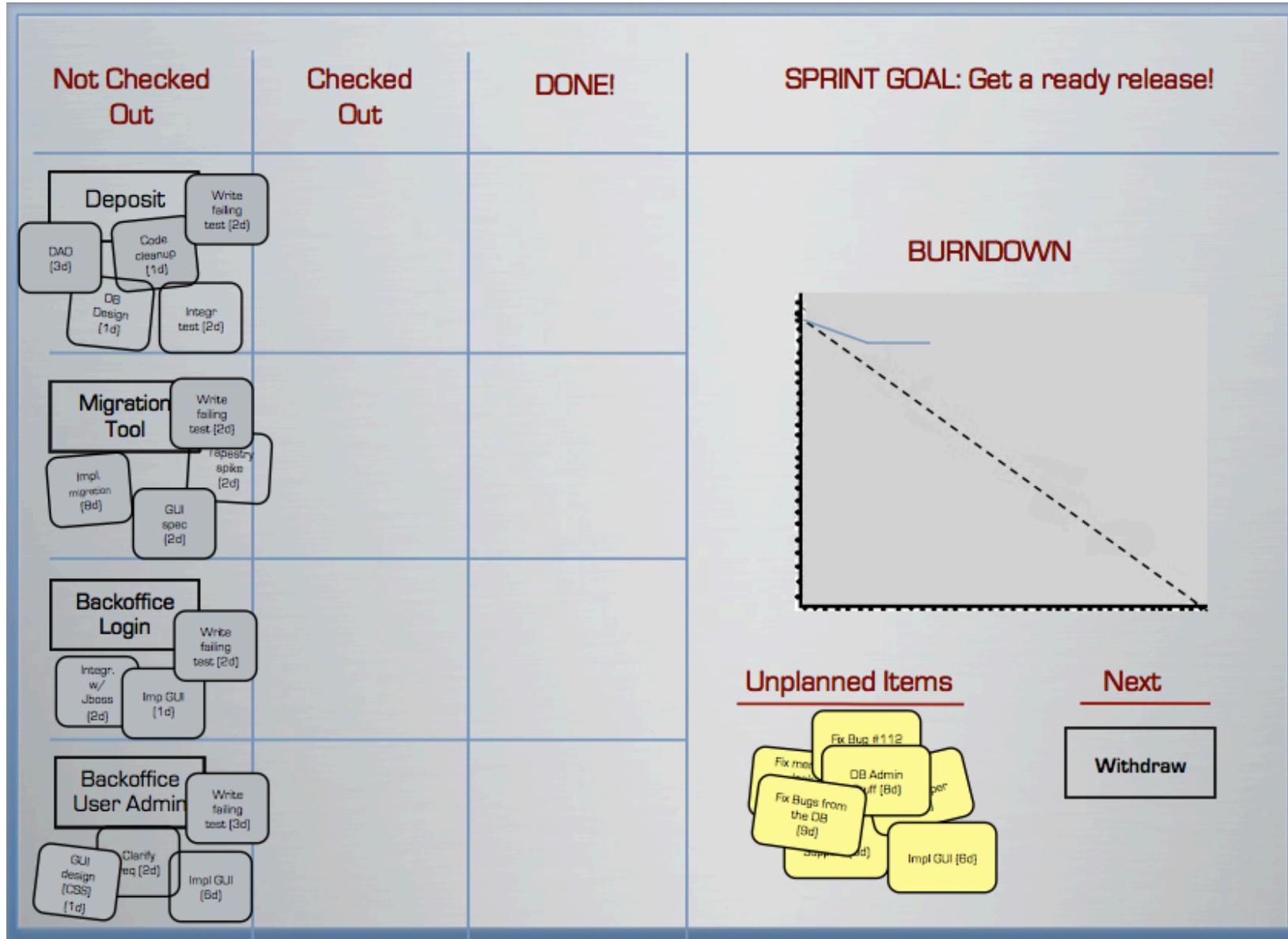
Warning sign #1



Source: Revised after Henrik Kniberg

© 1993-2012 Jeff Sutherland

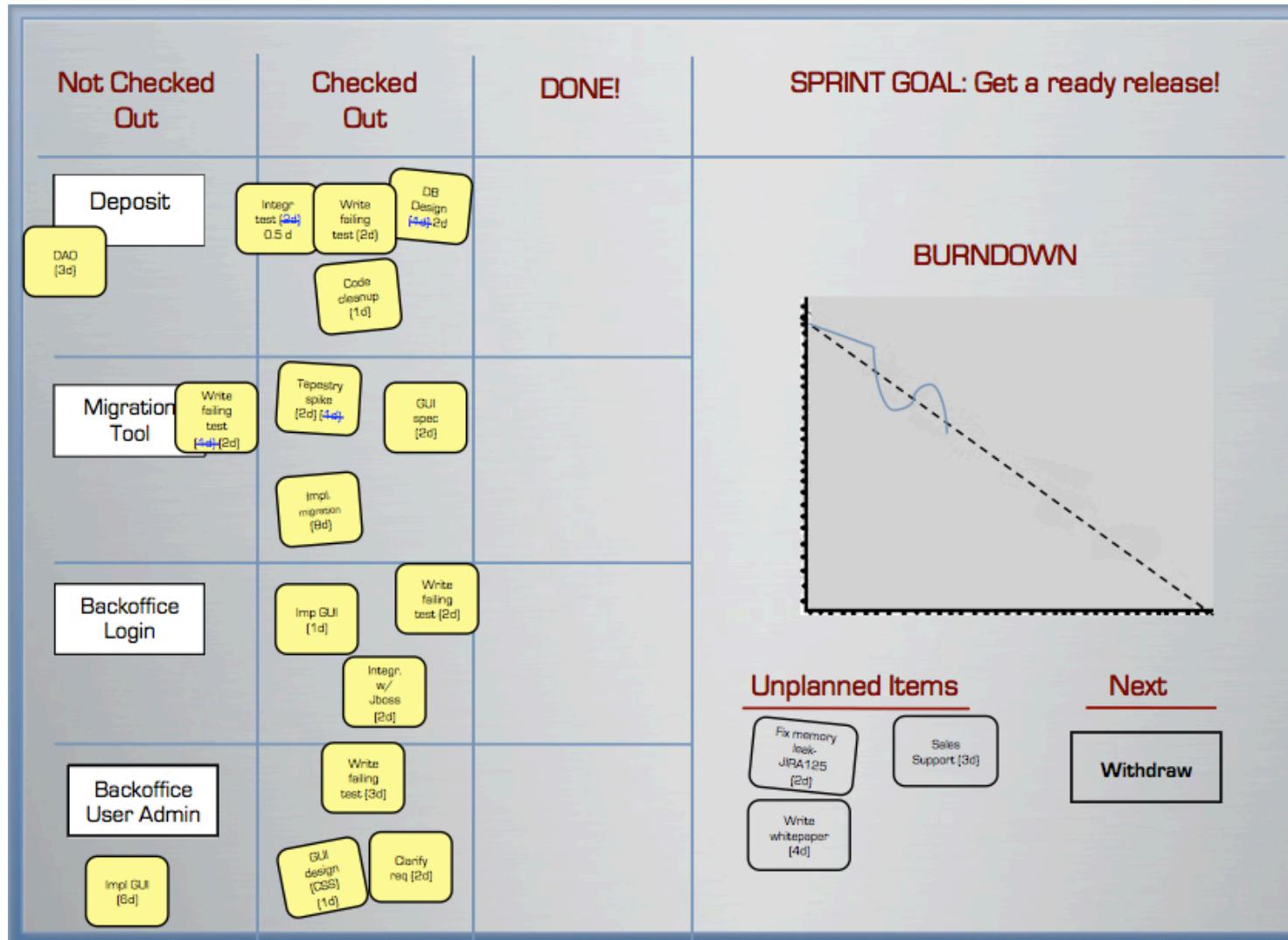
Warning sign #2



Source: Revised after Henrik Kniberg

© 1993-2012 Jeff Sutherland

Warning sign #3



Cost: 50000 Euro/mo/team

Source: Revised after Henrik Kniberg

© 1993-2012 Jeff Sutherland

WAIT A SEC

How is that burndown calculated?

Pattern: Scrum Emergency Procedure

When team is more than 20% behind,
execute the **Scrum Emergency Procedure**
by mid-Sprint:

- Innovate - do something different
 - Identify impediments, root cause analysis, remove impediments, otherwise ...
- Offload Sprint Backlog - get someone else to do it, otherwise ...
- Reduce scope in collaboration with Product Owner, or if this is not possible then ...
- Abort the Sprint
 - Recommended for new teams that need to learn how to do better estimates



buyinggoldcoins.net

© 1993-2012 Jeff Sutherland



Scrum Retrospective

Sprint Retrospective

3 roles

- Product owner
- Scrum master
- Team

3 artifacts

- Product backlog
- Sprint backlog
- Sprint burndown

3 activities

- Sprint planning
- Daily scrum
- Sprint review
 - Demo
 - Retrospective

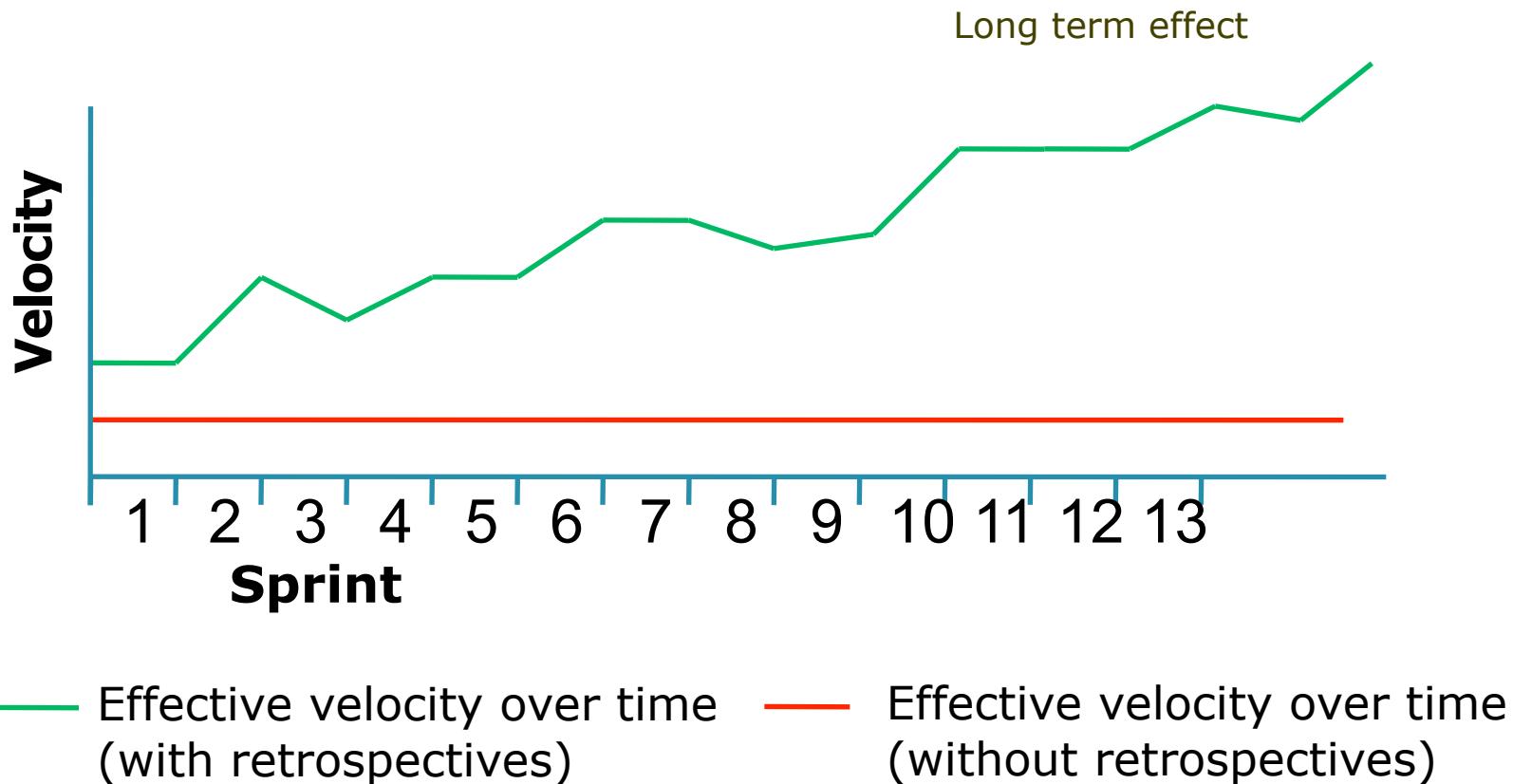


Source: Henrik Kniberg

“Rules”

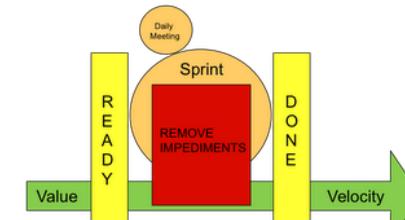
- Only address one impediment
- Put the kaizen in the backlog for the sprint - Scrum the Scrum
- Action should usually yield results quickly
- Communicate actions (and success or not) back to the Team
- And the hardest rule: Use common sense.

Sprint Retrospective



Source: Henrik Kniberg

Scrumming the Scrum



... you are using Scrum as a process improvement. The team must have effective [SPRINT RETROSPECTIVES](#) and have [POPPED THE HAPPY BUBBLE](#). The basic Scrum mechanisms are in place, and you want to leverage Scrum to fulfill its vision of kaizen:

kai-zen (カイゼン) *n. a Japanese business philosophy of continuous improvement of working practices, personal efficiency, etc. <ORIGIN> Japanese, literally ‘improvement’. The New Oxford American Dictionary.*

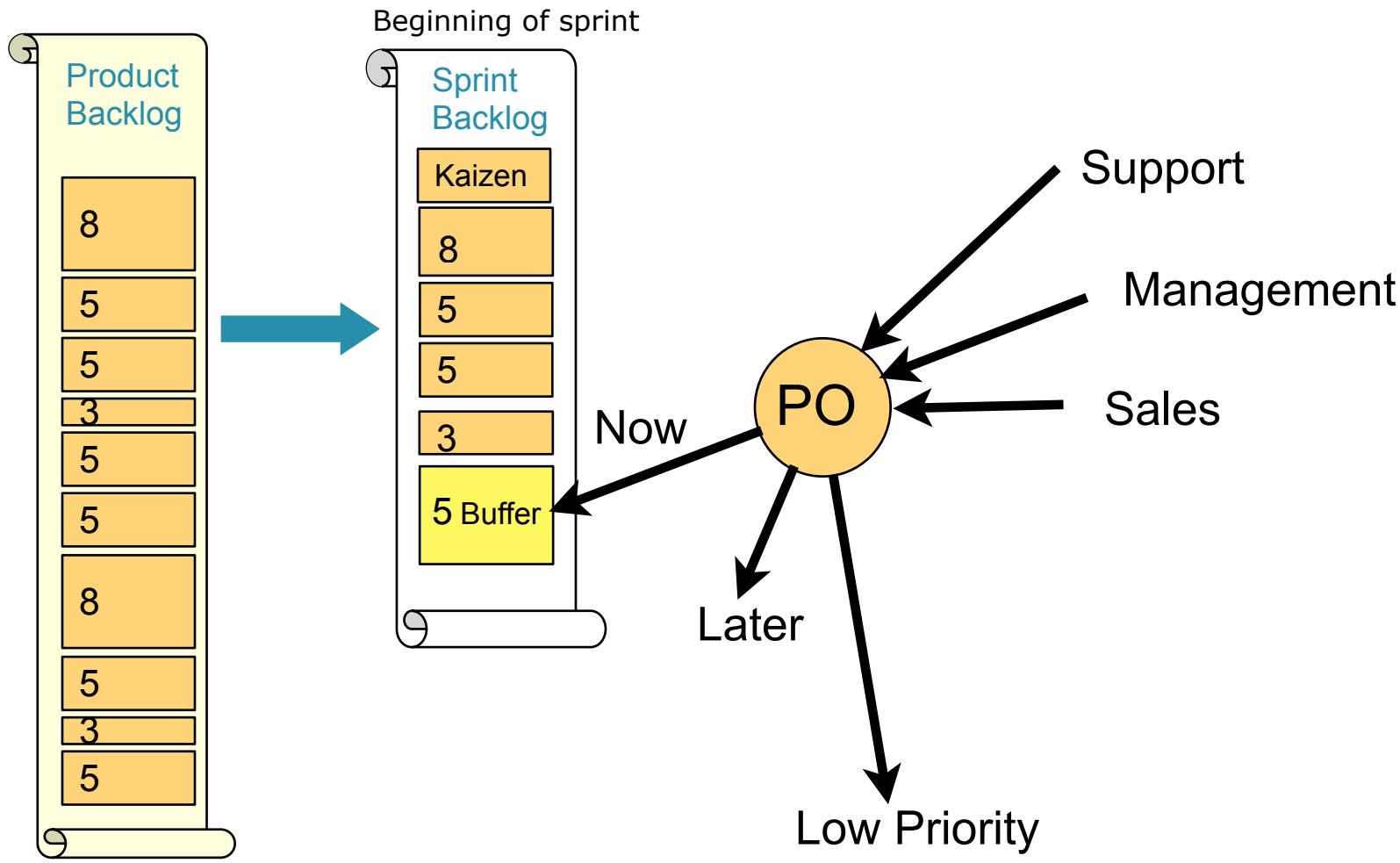
Only a small minority of Scrum teams achieve the hyperproductive state. This is because most teams fail to identify and remove impediments. Their software is not done, their backlog is not ready, and the team does not self-organize to improve performance.

Difficult impediments require extreme focus to remove. Working on many impediments at once often leads to a lot of work with little gain and can demoralize the team.

Therefore: *Identify the single most important impediment at the Sprint Retrospective and remove it before the end of the next sprint.*

To remove the top priority impediment, put it in the [Sprint Backlog](#) as a user story with acceptance tests that will determine when it is Done. Then evaluate the state of the story in the [Sprint Review](#) like any other task.

Pattern: Illigitimus Non Interruptus



Points vs. Hours

- Rand Corporation received a grant from U.S. DOD in the 1940's to determine best way to estimate tough projects
 - Discovered estimation in hours has high error rate and wide variance
 - Found people could put things in relative size piles best
- Fibonacci growth pattern easiest for humans
 - Seen everywhere in nature
 - RAND called it the Delphi technique