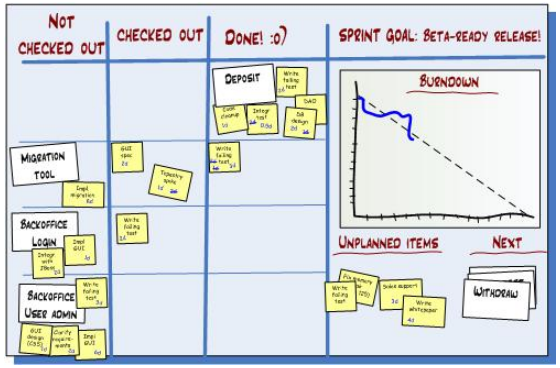
Scrum intro

1. Create Name Tag



2. In groups of three, discuss what you know about agile and Scrum

crisp

About me

telenor

Polisen

Skatteverket

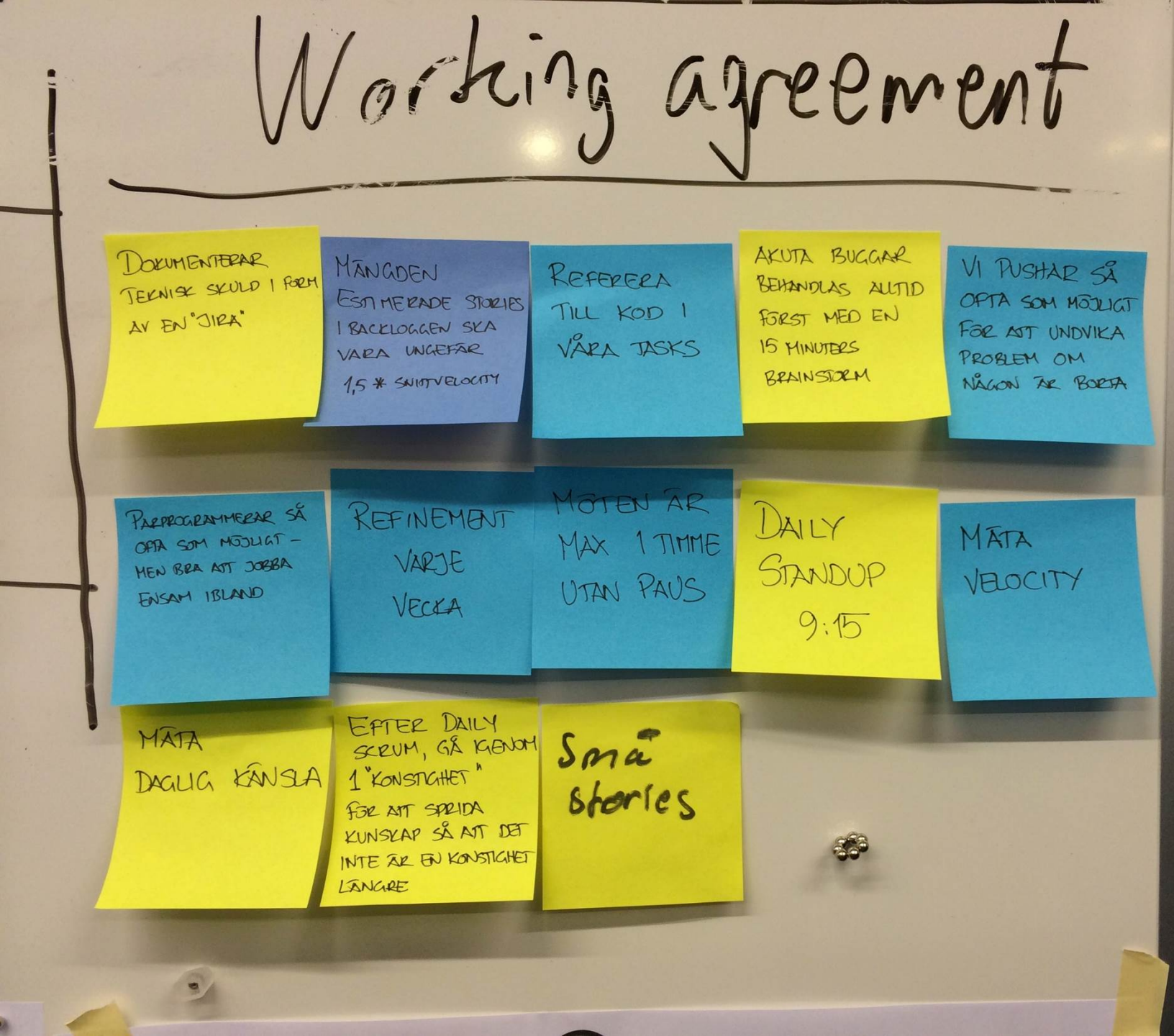
Vår vision är ett samhälle där alla vill göra rätt för sig

- 20 years being Developer/SW-Architect
- Team coach/Agile coach
- Teacher
- mikael.brodd@crisp.se
- www.crisp.se/mikaelbrodd
- @mibrodd

crisp

Working agreement

- Phone/SMS
- E-mail/other computer stuff
- Lunch


Working agreement
Dokumenterar teknisk skuld i form av en "JIRA"
Mängden Estimerade stories i backloggen ska vara ungefär 1,5 * snittvelocity
Referera till kod i våra jasks
Akuta buggar behandlas alltid först med en 15 minuters brainstorm
Vi pushar så ofta som möjligt för att undvika problem om någon är borta
Parprogrammerar så ofta som möjligt - men bra att jobba ensam ibland
Refinement varje vecka
Möten är max 1 timme utan paus
Daily Standup 9:15
Mäta velocity
Mäta daglig känsla
Efter daily scrum, gå igenom 1 "konstighet" för att sprida kunskap så att det inte är en konstighet längre
Små stories

Good to know before we start

- Slides www.crisp.se/scania
- Disclaimer!

Vision for this course

Understand the basics of Scrum.

Artifacts
Systems thinking
Roles
Agile
Scrum
Practices
Principles
History
Planning
Theory...

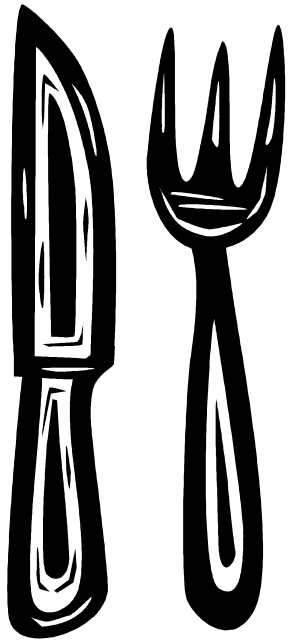
crisp

Hopes, questions and expectations

- **Take two minutes to write down on post-it notes**
  - Hopes and wishes about this Course
  - Questions you want to have answered
- **One item per post-it**

Any tool can be misused

Compare tools for understanding, not judgement

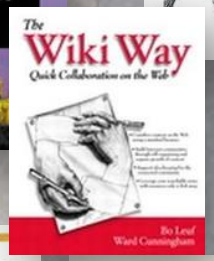


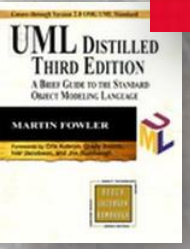
Don’t blame the tool!

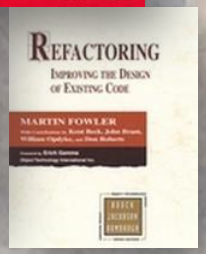
crisp

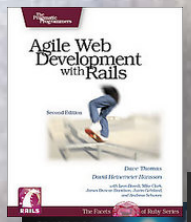
Roots of Agile

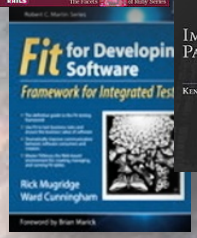
Agile Manifesto

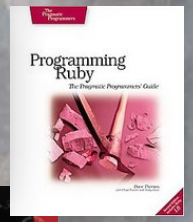


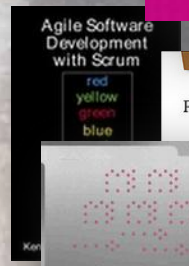


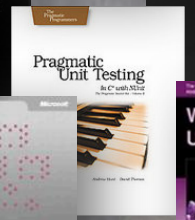


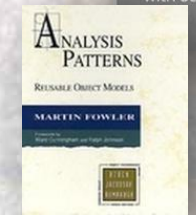












www.agilemanifesto.org

We are uncovering better ways of developing software by doing it and helping others do it.

Feb 11-13, 2001

Snowbird ski resort, Utah

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt

Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

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**

Individer och interaktioner framför processer och verktyg

Working software over **comprehensive documentation**

Fungerande programvara framför omfattande dokumentation

Customer collaboration over **contract negotiation**

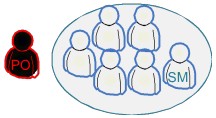
Kundsamarbete framför kontraktsförhandling

Responding to change over **following a plan**

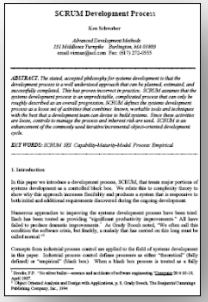
Anpassning till förändring framför att följa en plan

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

History

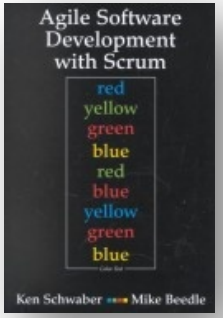


1993
First Scrum team formed by Jeff Sutherland at Easel Corp.


SCRUM Development Process

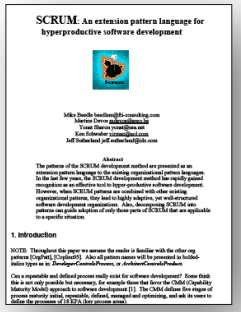
Ken Schwaber

1998
”Scrum, a pattern language for hyperproductive software development” published by Ken, Jeff, et al.



2001
”Agile Software Development with Scrum” published by Ken Schwaber and Mike Beedle.

1996
”Scrum Development Process” published by Ken Schwaber.


SCRUM: An extension pattern language for hyperproductive software development

2001
Agile manifesto creation

crisp

The Scrum Guide

The Definitive Guide to Scrum:
The Rules of the Game

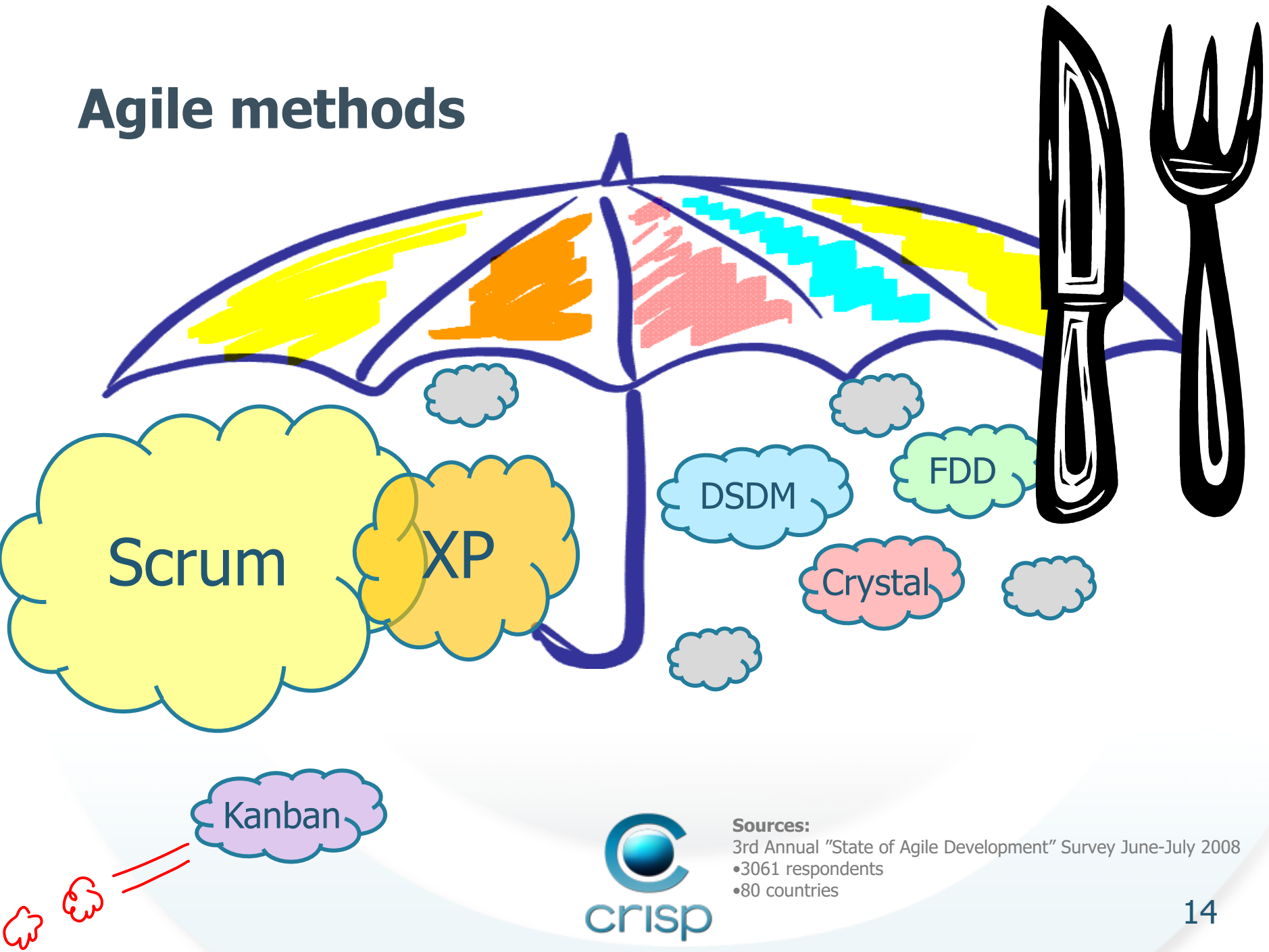
October 2011

Developed and sustained by Ken Schwaber and Jeff Sutherland

Scrum.org

Improving The Profession Of Software Development

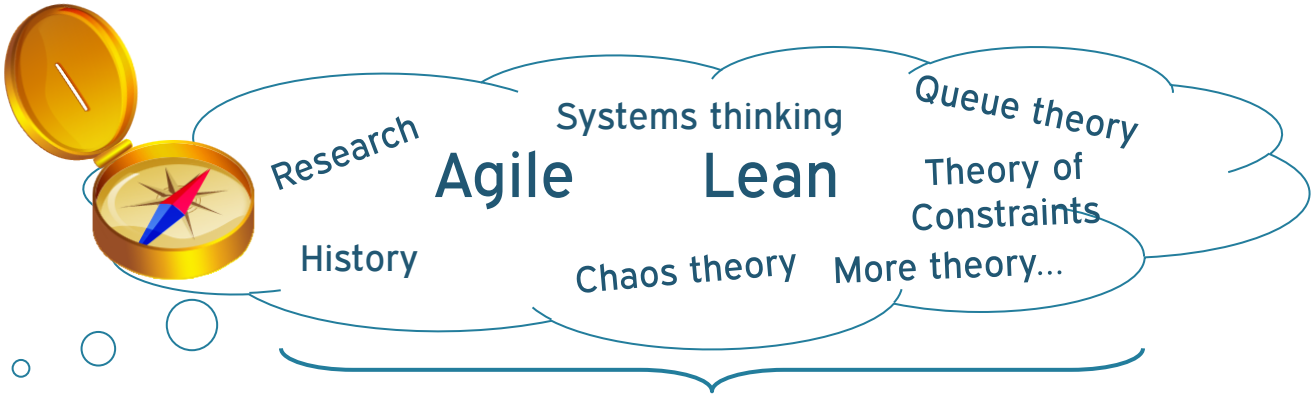
Agile methods



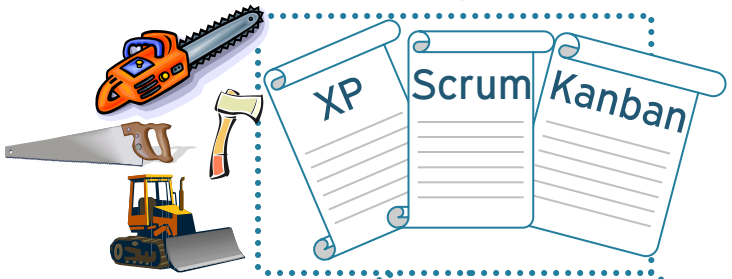
Sources:
3rd Annual "State of Agile Development" Survey June-July 2008
•3061 respondents
•80 countries

crisp

The Big Picture

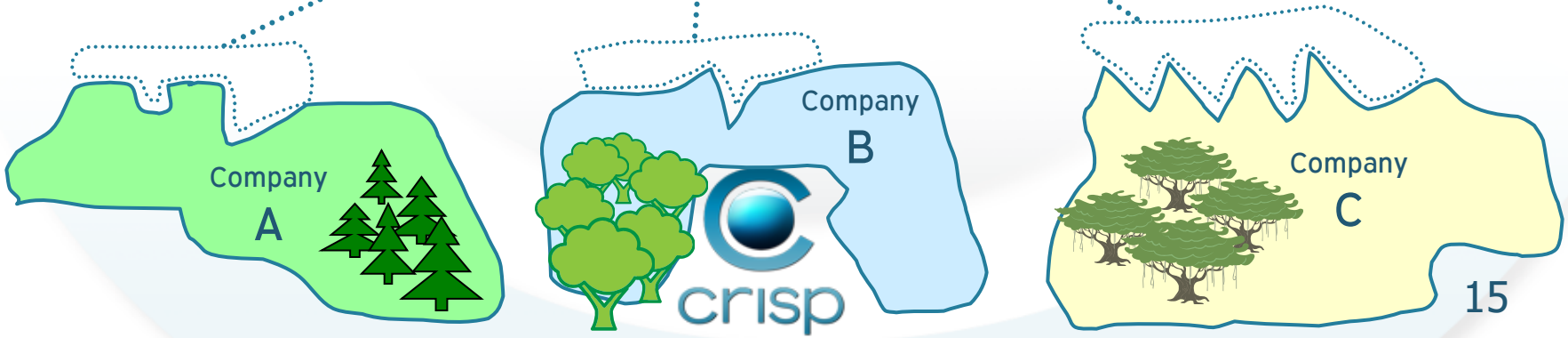


Principles



Practices

Implementation



Do we really need to change culture?

More powerful & sustainable more change pain

- Mobilizing brainpower
- Fast feedback
- Empirical development

Values

Requires a deep commitment

- Direct communication
- Zero bug policy
- Empowered people

Principles

Requires structural and cultural changes

- Test Driven Dev.
- Beyond budgeting
- Agile HR practices

Practices

- Scrum
- Story mapping
- Scrum board

Tools and process

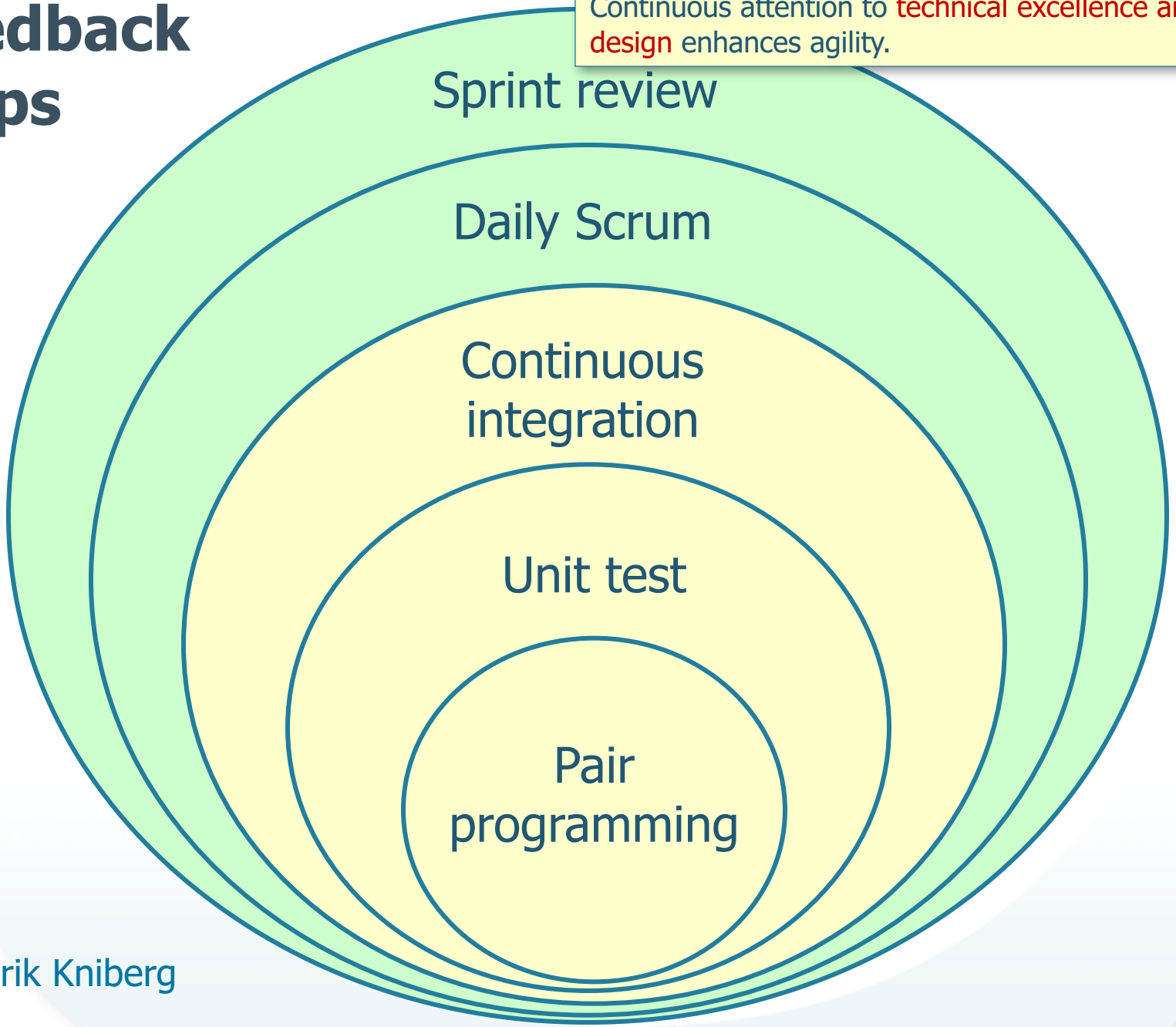
Can be adopted in command and control culture

More visible, less powerful, low change pain

crisp

Feedback loops

Principle #9:
Continuous attention to technical excellence and good design enhances agility.



Henrik Kniberg

Scrum Overview

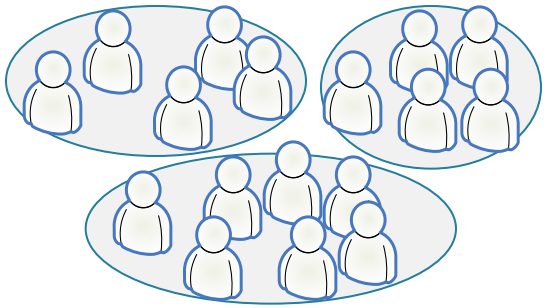
Why it's called Scrum

The… 'relay race' approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."

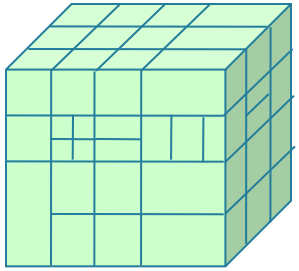
Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game", *Harvard Business Review*, January 1986

Scrum in a nutshell

Split your organization

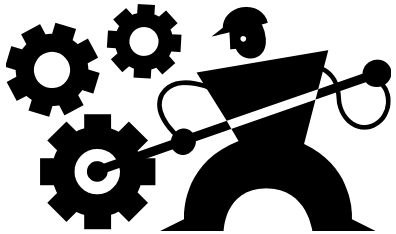


Split your product

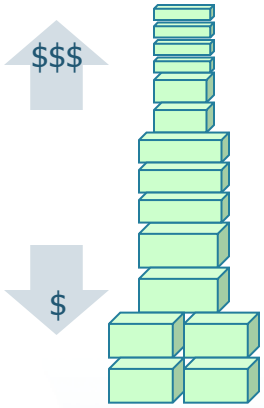


~~Large group~~ spending ~~a long time~~ building a ~~big thing~~
Small team spending a little time building small thing
... but integrating regularly to see the whole

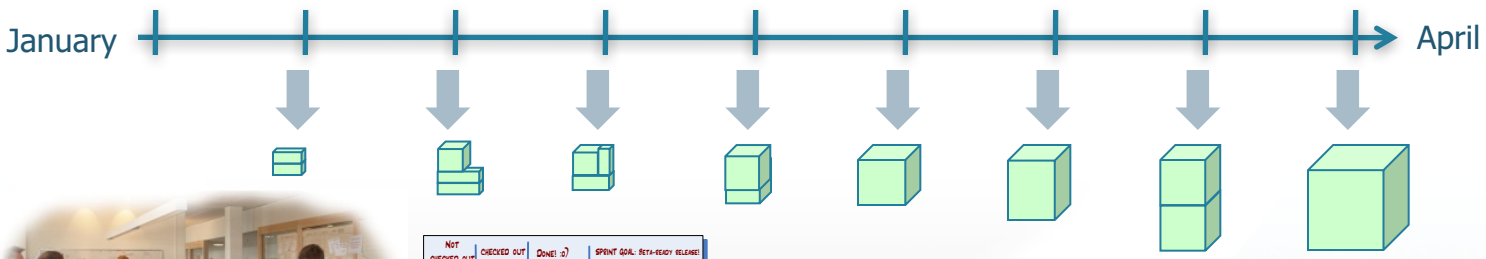
Optimize process

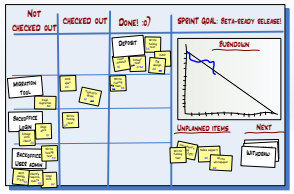


Optimize business value



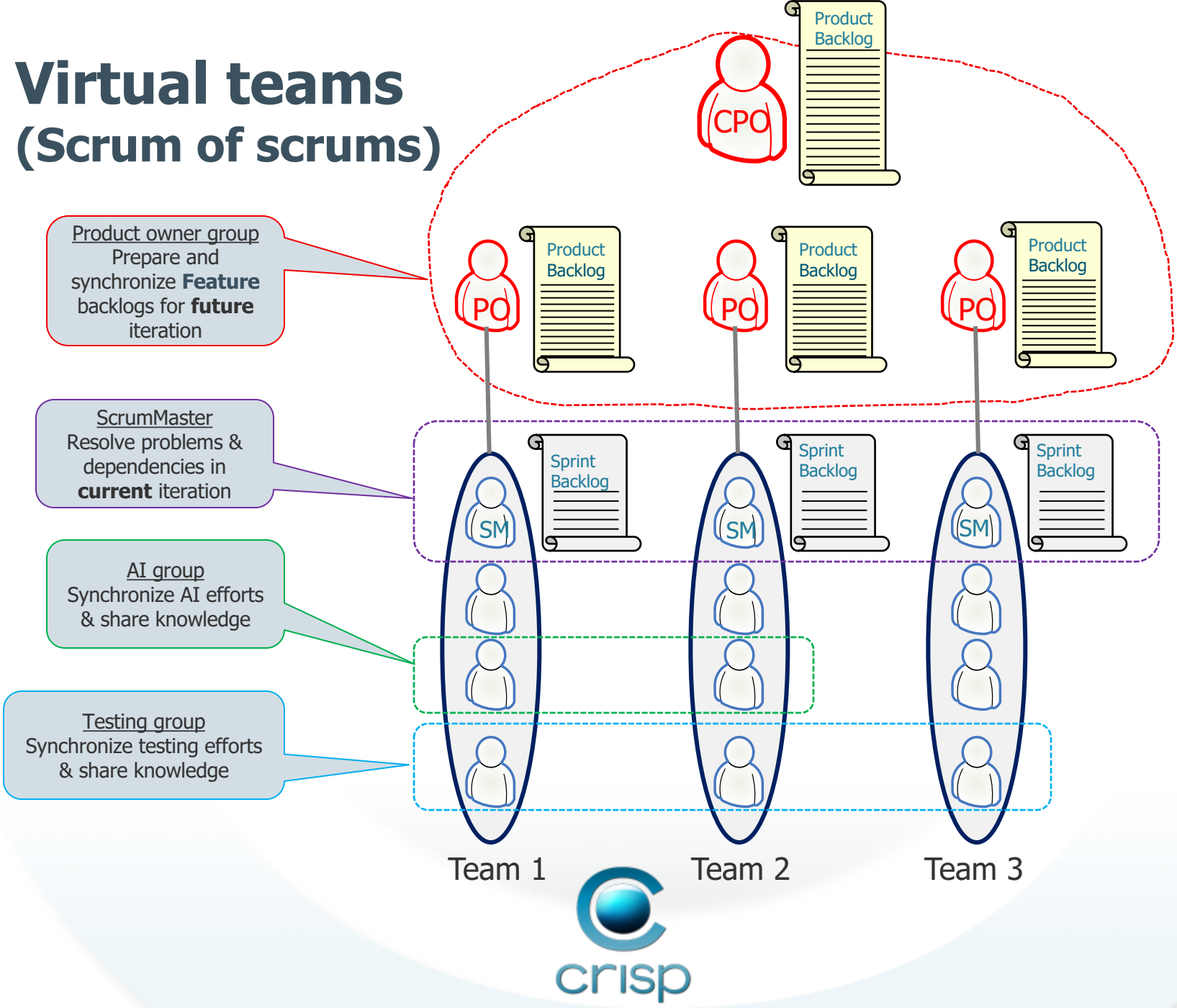
Split time



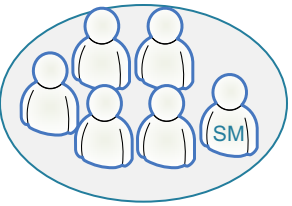


crisp

Virtual teams
(Scrum of scrums)

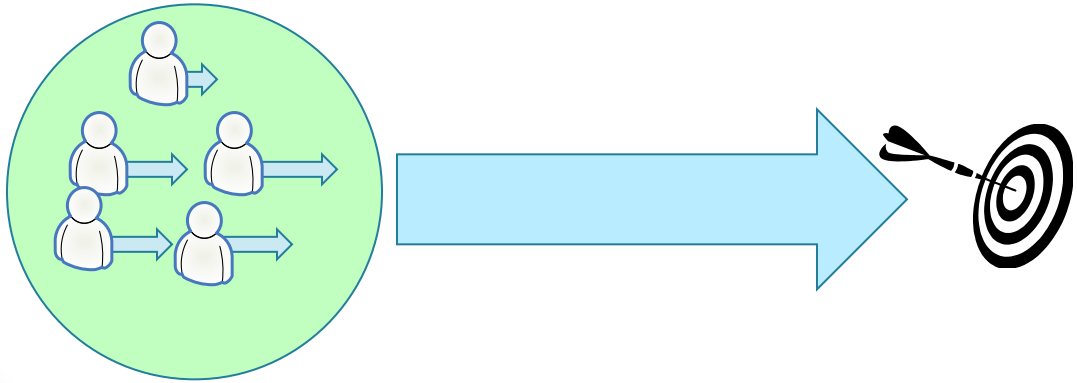


crisp



Team

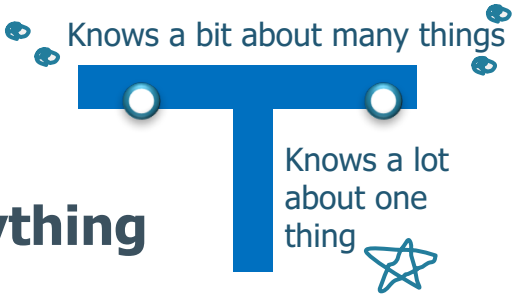
- 3 – 8 full-time individuals
- Cross-functional
- Sits together
- Shared responsibility
- Self-organizing



crisp

Cross functional team

Doesn't mean everyone has to know everything



Product Backlog

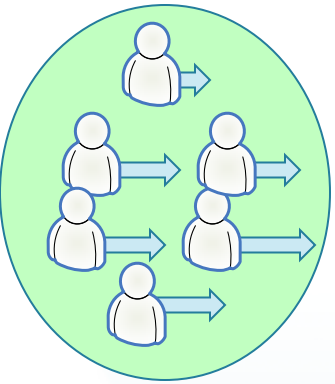
I *can* test, but I'm not so good at it.

Skills Needed to implement Top X backlog items

	Test	DB	Web	Java	Domain	CM
Lisa	○	•	○	☆		•
Joe	•	☆		•	•	
Fred	•			☆	•	○
Jenny	•		☆	○	○	•
David	☆			•	☆	
Erik	○		☆	☆	•	☆

I'm good at Java!

I don't know CM at all. But I'm willing to learn!



I won't even go near a database!

crisp

ScrumMaster

- **Enforces Scrum practices**
  - **Coaching** rather than command & control
- **Protect the team**
- **Removes impediments**
- **Usually part of the team**

- **Usually Not the line manager**
- **Usually Not the tech guru**

Impediment backlog

- Slow workstations
- Interference from sales
- No test environment
- No contact with customer
- Crowded office

Is ScrumMaster a full-time role?

	Small team	Large team
Few problems	≈ 10%	≈ 50%
Many problems	≈ 50%	100%

crisp

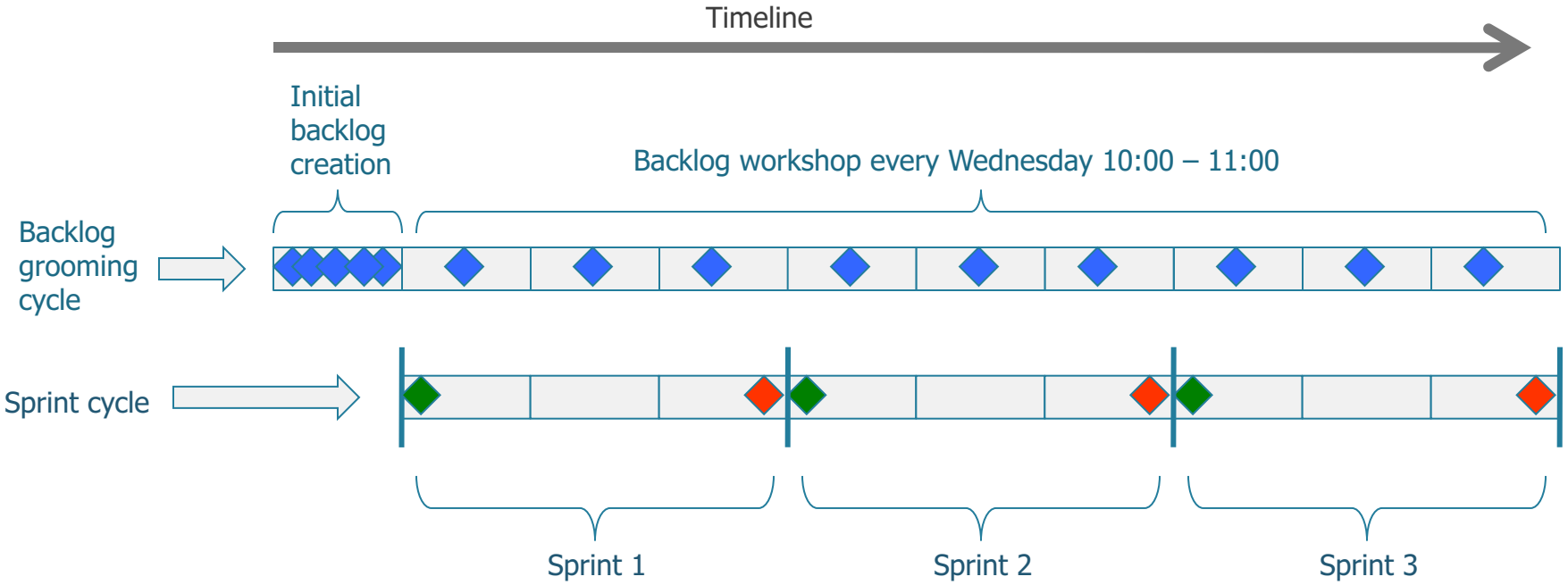
Product owner

- **Represents all stakeholders**
  - Available for the team
- **Decides where the team should go**
  - Not how they get there
  - Not their speed
- **Defines scope / vision / roadmap**
  - Domain knowledge
- **Owns product backlog**
- **Prioritizes**
  - Mandate to prioritize

- **Does not estimate size of stories**
- **Usually Not the line manager**

crisp

Backlog creation & workshop – sample schedule



Product Owner

Owns the vision: Where are we going and why?	Orders the User Stories in the Backlog
Owns the Product Backlog (i.e. makes sure it exists and is, is in good shape, and up to date)	Responsible for Release Planning and Road Mapping (i.e. dates, scope and tradeoffs)
Enables direct communication between team and stakeholders	Has authority to cancel a sprint (for example if the sprint goals become obsolete)

ScrumMaster

Enforces Scrum values and practices

Removes (and helps the team resolve) barriers and impediments

Shields the team from external interferences

Coaches the team to high performance (catalyst for improvements)

Teaches agile practices

Helps team visualize work, progress, velocity and capacity

Development Team

Estimates User Stories
(i.e. work effort)

Self-organizes
(Decides who does what in which order, i.e. owns the teams process)

Decides scope of the sprint
(i.e. how much work to pull in)

Cross-functional
(i.e. has all skills necessary to build and deliver)

Responsible for quality of
what is delivered

Decides how to design and build (from a technical perspective)

Undefined role

Sets salaries

Collects time reports

Conducts Performance Reviews

Decides how teams are put together

Sprint Planning

Team decides how many User Stories to pull into the sprint based on previous velocity (and current capacity)

Define the sprint's goal (i.e. primary achievement)

Team breaks down the work needed to fulfill User Stories into tasks

Daily Stand-up

Team decides what they need to do today to get closer to the sprint's goal

Problems and impediments are raised and addressed

If progress towards the sprint's goal aren't satisfactory team decides upon actions (what to do) and alerts Product Owner

Backlog Refinement

Big User Stories are split into smaller User Stories (and re-estimated)

New User Stories are clarified and estimated

Top User Stories in the Product Backlog are made clear enough to be pulled into next sprint

Sprint Review

Finished User Stories are demonstrated for Product Owner and Stakeholders. Feedback is collected.

Review of results and summary of sprint (What was finished? Delivered? Problems resolved? Interfering blockers?)

Review of Product Backlog, budgets, Roadmap, release dates, etc.

Sprint Retrospective

Inspection of how the last sprint went (with regards to people, process and tools)

Identification and agreement upon potential process improvements

Creation of plan for implementing identified process improvements (actions points)

Exercise: Ball game

Exercise – Ball game

Goal: Collect as many points as possible during 2 minutes by passing as many balls as possible

- Same ball must be touched by everyone in your team before it counts as 1 points
- Ball must have air time when passed between everyone (can't be handed over)
- Can't pass to your direct neighbour
- Use nothing but hands

Sprint	Est	Act	Changes after sprint
1			
2			
3			

Alt 1) Work – Improve – Work - Improve

Alt 2) Just work

Alt 3) All planning upfront

Deming Cycle

Lean - Kaizen

Plan	Do	Check	Act

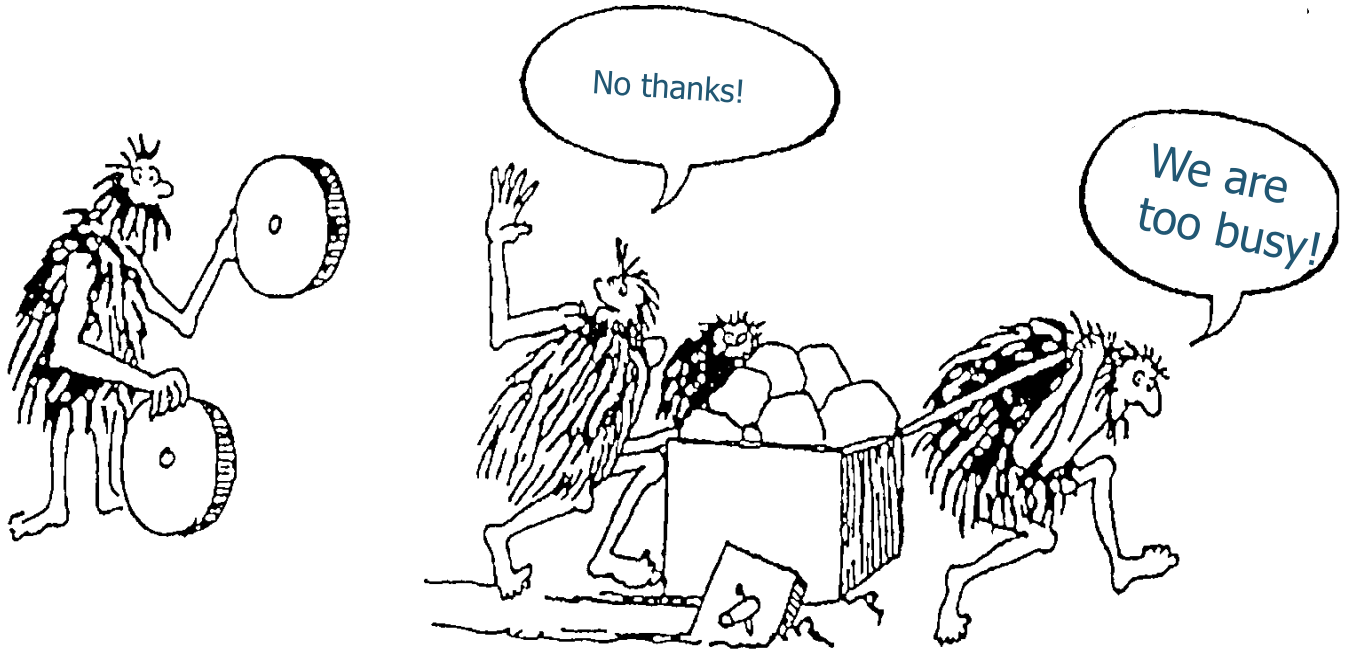
Scrum

Sprint Plan	Sprint	Review	Retro-spective

Scientific Method

Hypo-thesis	Test	Observation And Inference	Confirm, Modify, Discard

By Dr. W. Edwards Deming, who is considered by many to be the father of modern quality control


No thanks!
We are too busy!

Perfection is a direction, not a place

Empirical Product Development

Vill alla göra användbara system?

Intervju med en IT-chef:

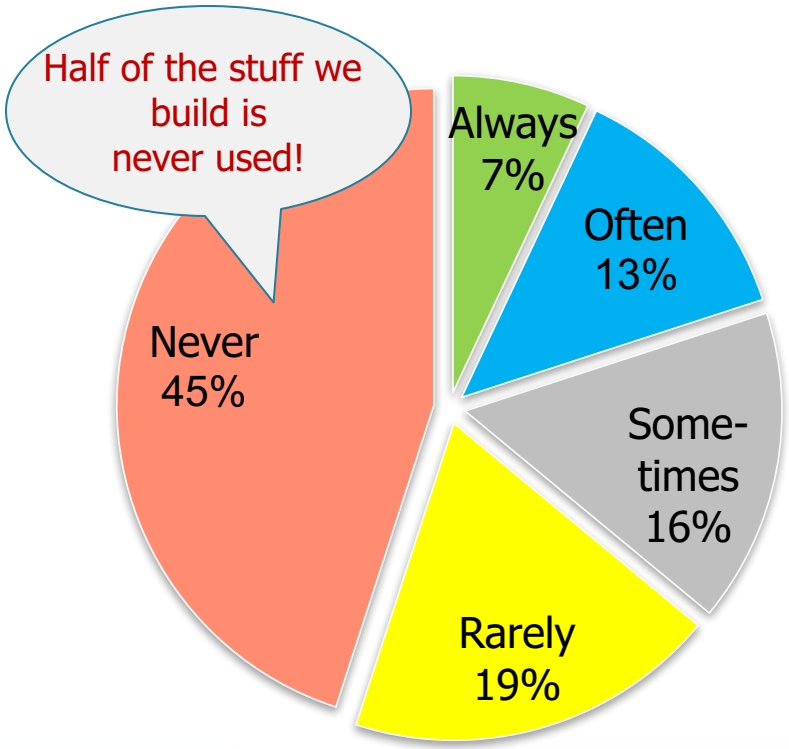
- *Bygger ni användbara system?*
- *Nej.*
- *Varför inte det?*
- *Beställaren beställer inte det.*
- *Skulle ni inte kunna tänka er att bygga användbara system, även om beställaren inte beställt det?*
- *Nej, för om vi ska göra det måste vi blanda in användaren, och det kommer att ge oss allehanda organisatoriska problem och det intresserar inte mig. Som IT-chef är min roll att leverera det beställaren beställer, varken mer eller mindre, på kortast möjliga tid med utnyttjande av så lite resurser som möjligt. Det är då jag har gjort ett bra jobb!*

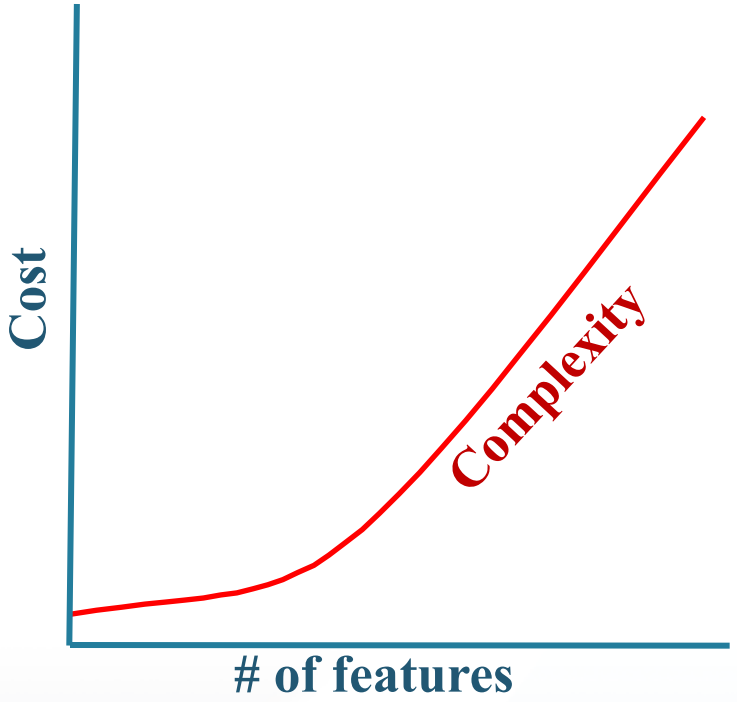
wedish Committee for Digitization

STATENS OFFENTLIGA UTREDNINGAR

We tend to build the wrong thing

Features and functions used in a typical system

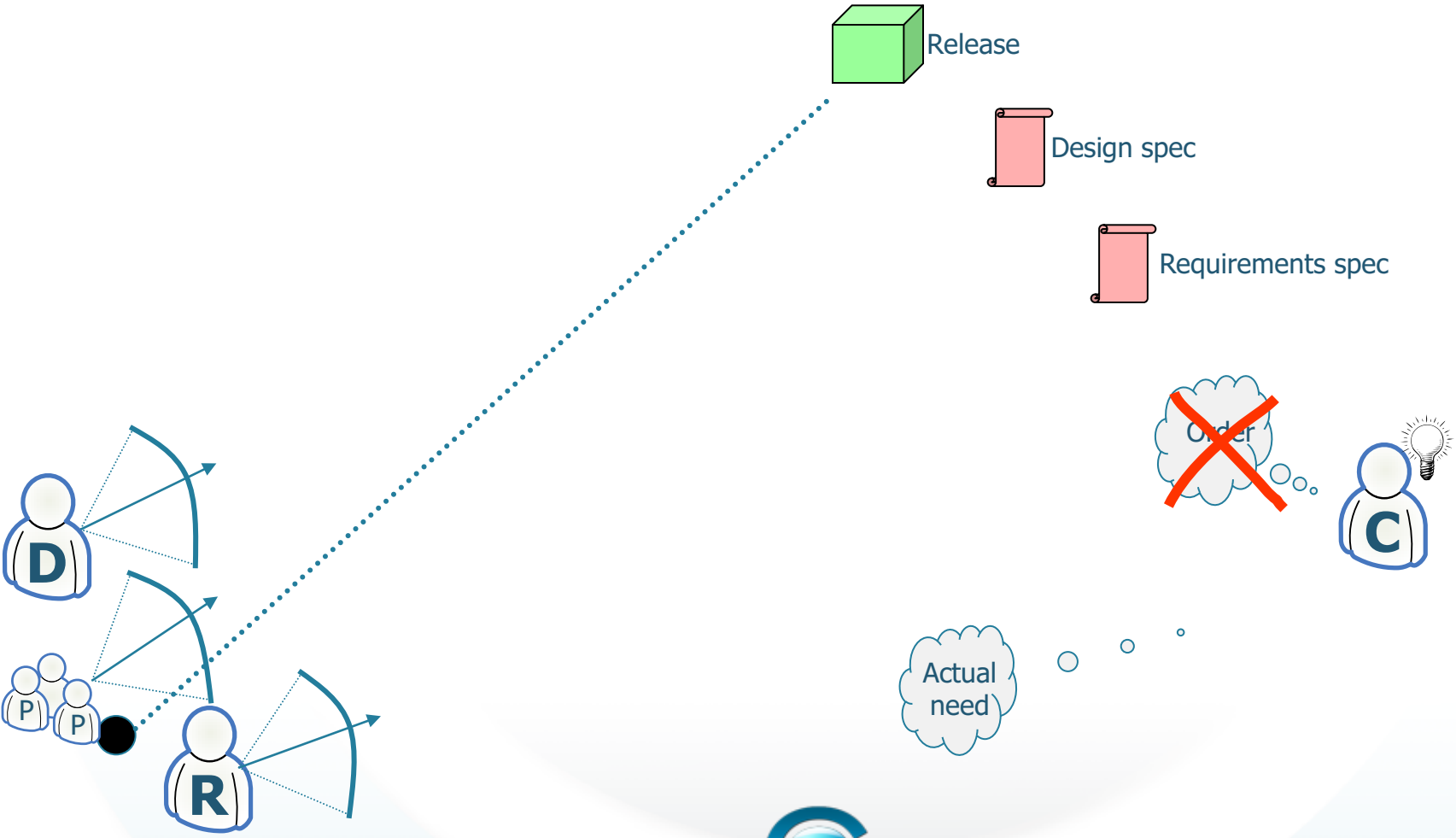




Sources:
Standish group study reported at XP2002 by Jim Johnson, Chairman

This graph courtesy of Mary Poppendieck

Predictive approach



crisp

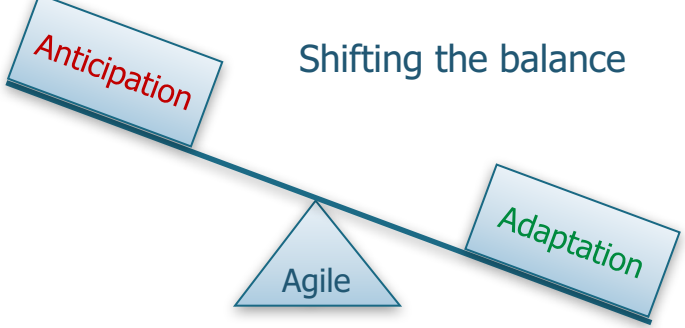
3 things we wish were true

- The customer knows what he or she wants
- The developers know how to build it
- Nothing will change along the way

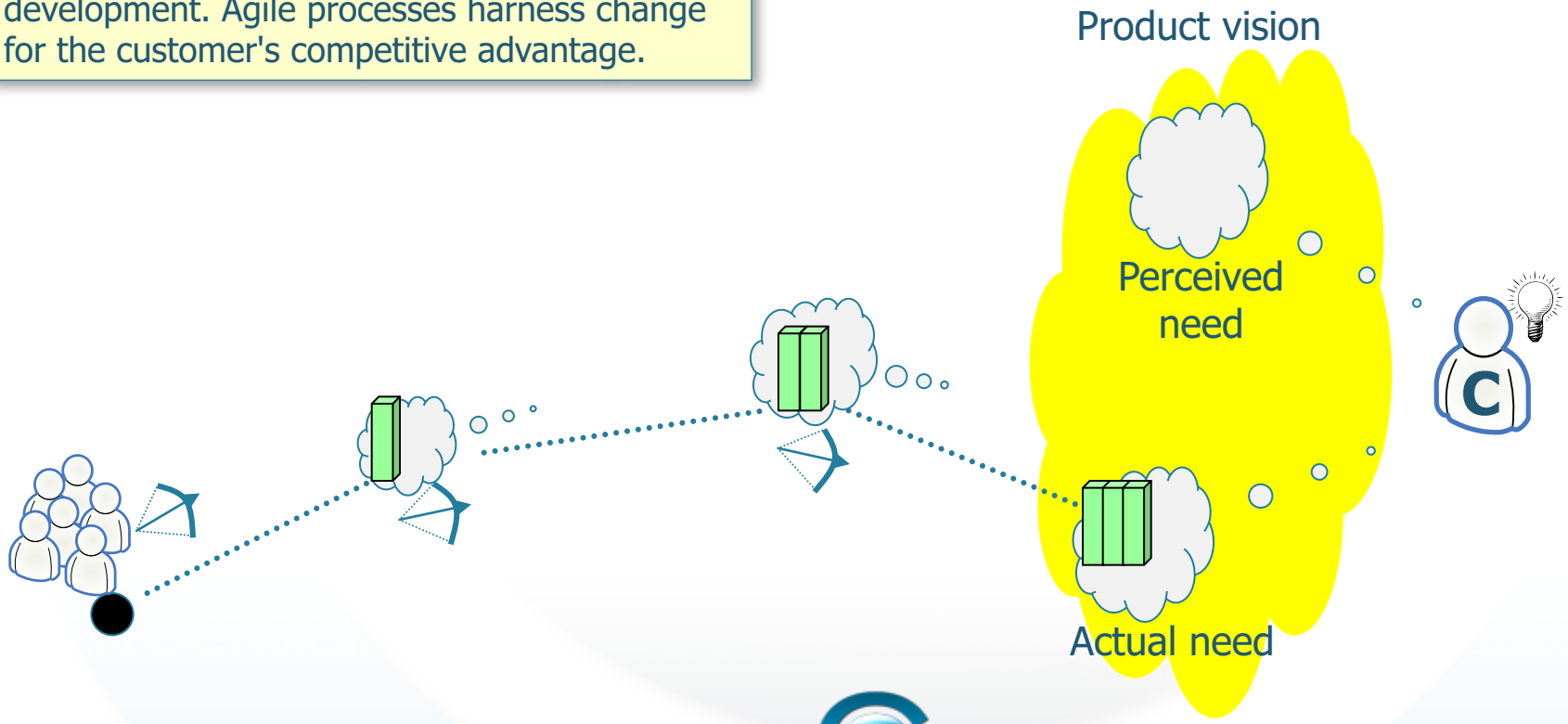
3 things we have to live with

- The customer discovers what he or she really wants
- The developers discover how to build it
- Many things change along the way

Adaptive (agile) approach



Principle #2:
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

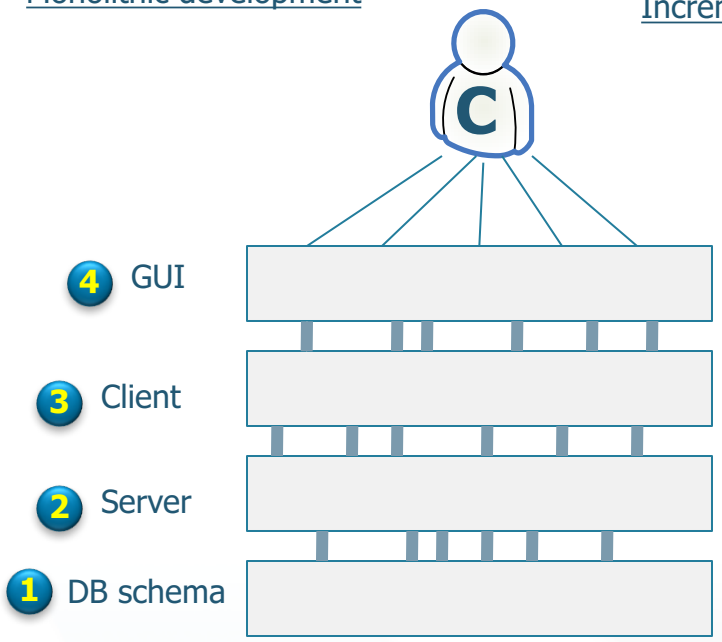


crisp

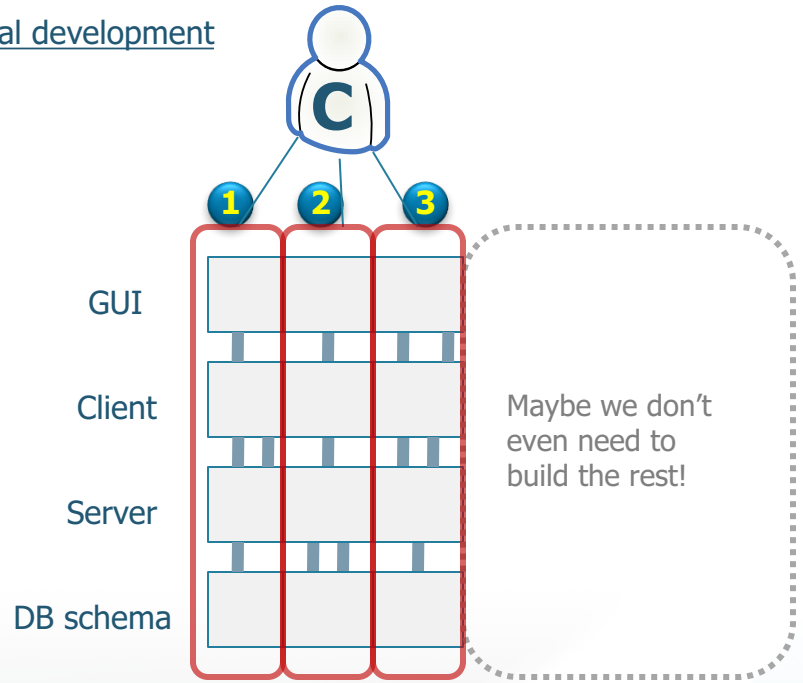
Iterative, incremental development

Iterative = don't expect to get it all right the first time
Incremental = build in "vertical" slices (features) rather than "horizontal" (layers)

Monolithic development

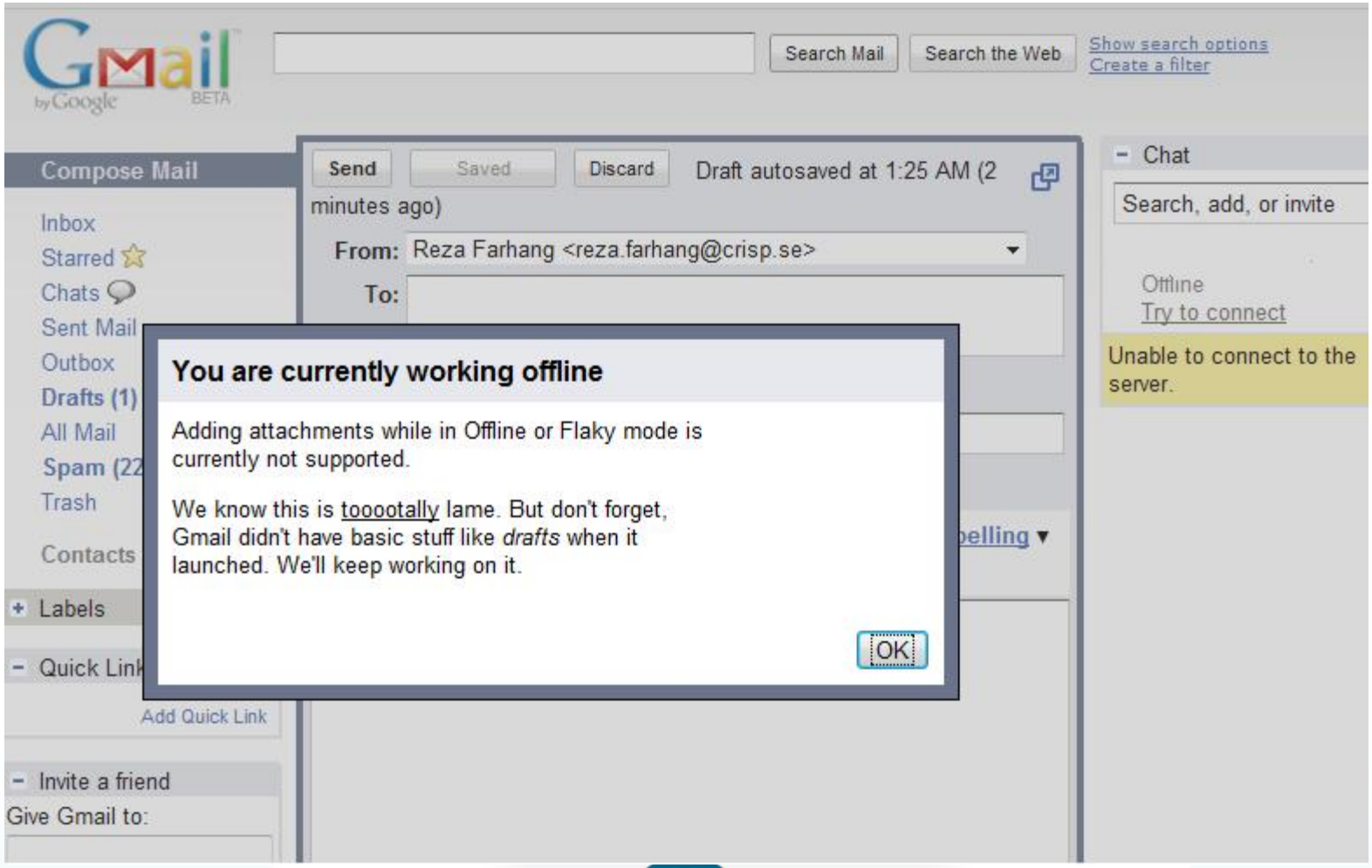


Incremental development



crisp

Incremental development



crisp

Successful company launching (too) early

2007

iPhone

2G
No GPS
No App-store
No MMS

2008

iPhone 3G

3G
GPS
App-store
No MMS

2009

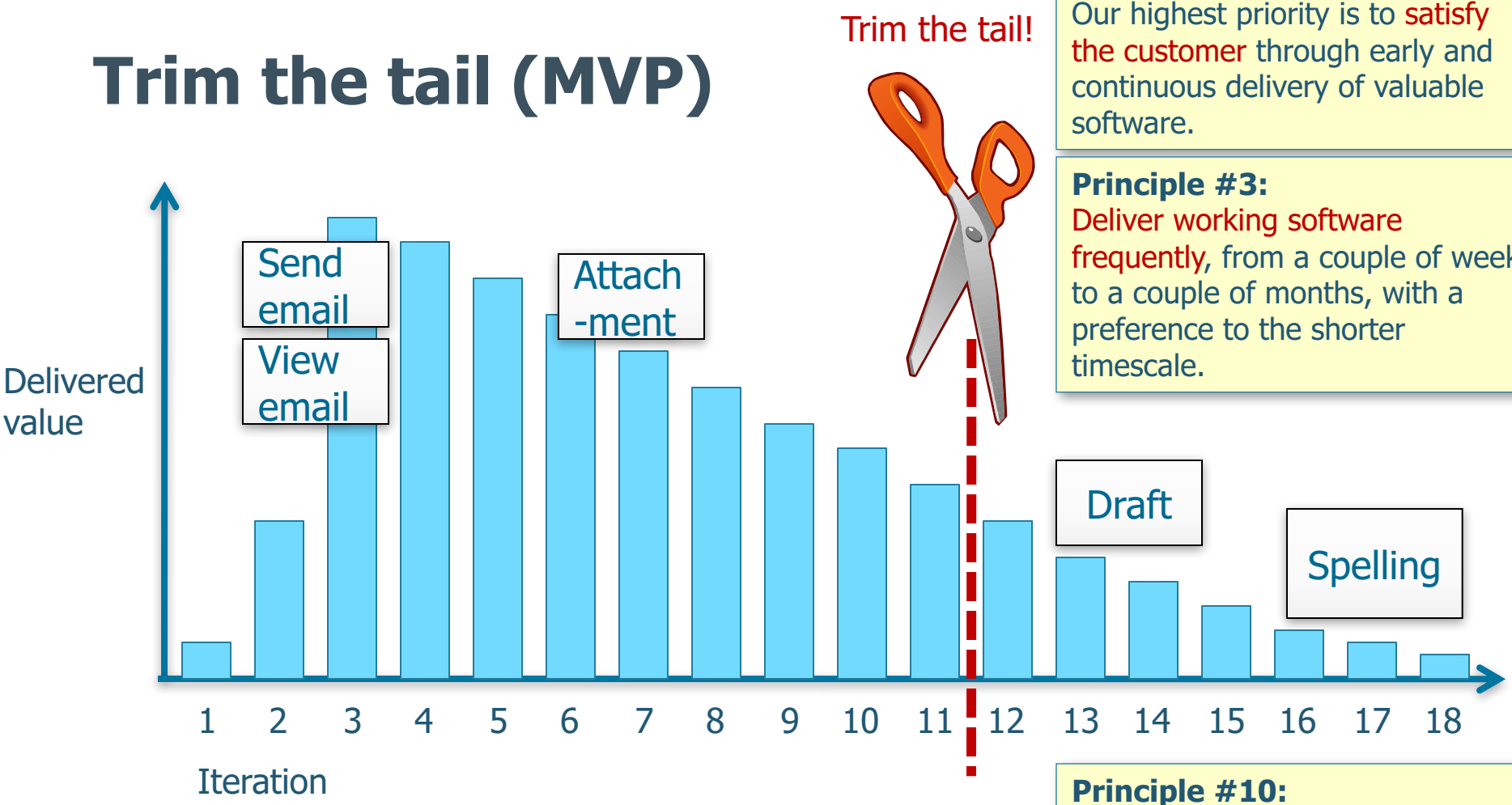
iPhone 3GS

MMS
Compass
Gyro

crisp

Maximize Value, not Output

Trim the tail (MVP)



Principle #1:
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Principle #3:
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Principle #10:
Simplicity--the art of maximizing the amount of work not done--is essential.

crisp

Product backlog & User Stories

Exercise - requirements

Product backlog

As a <stakeholder>
I want <what>
so that <why>

Product vision

Product Backlog

As a buyer
I want to save my shopping cart
so that I can continue shopping later

As a booker
I want to receive notifications when new available slots appear in the calendar
so that I don't have to keep checking manually

(... etc ...)

Default Definition of Done

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

= I haven't messed up the codebase

crisp

User story

As a <stakeholder>
I want <what>
so that <why>

As a buyer
I want to save my shopping cart
so that I can continue shopping later

2

How to demo:
1)Enter store
2)Put a book in shopping cart
3)Press ”save cart”
4)Leave store, and enter it again
5)Check that the book is in my cart

Independent
Negotiable
Valuable
Estimable
Small
Testable

Acronym courtesy of Bill Wake – www.xp123.com

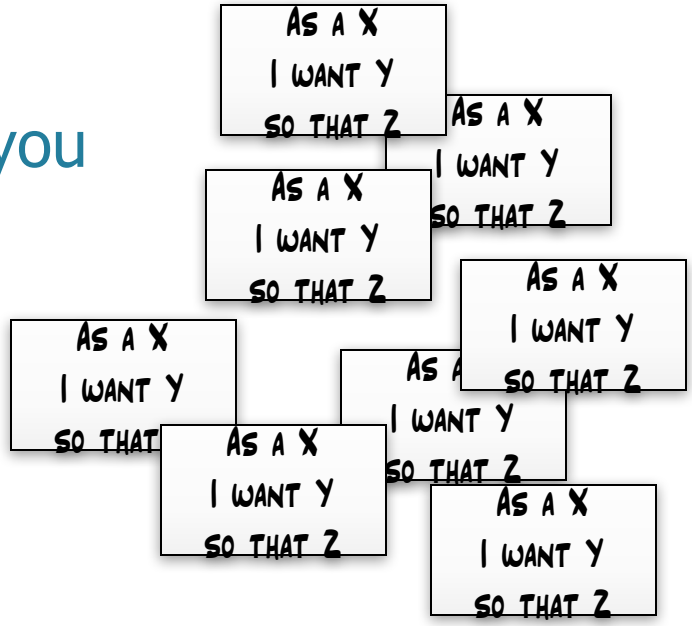
Exercise step 3

Write on index cards.
Use a thick pen.
Use the story template.

Create product backlog

- Write as many user stories as you can

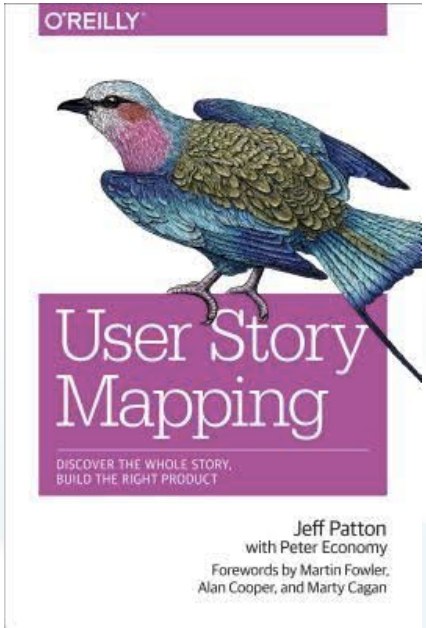
As a <stakeholder>
I want <what>
so that <why>

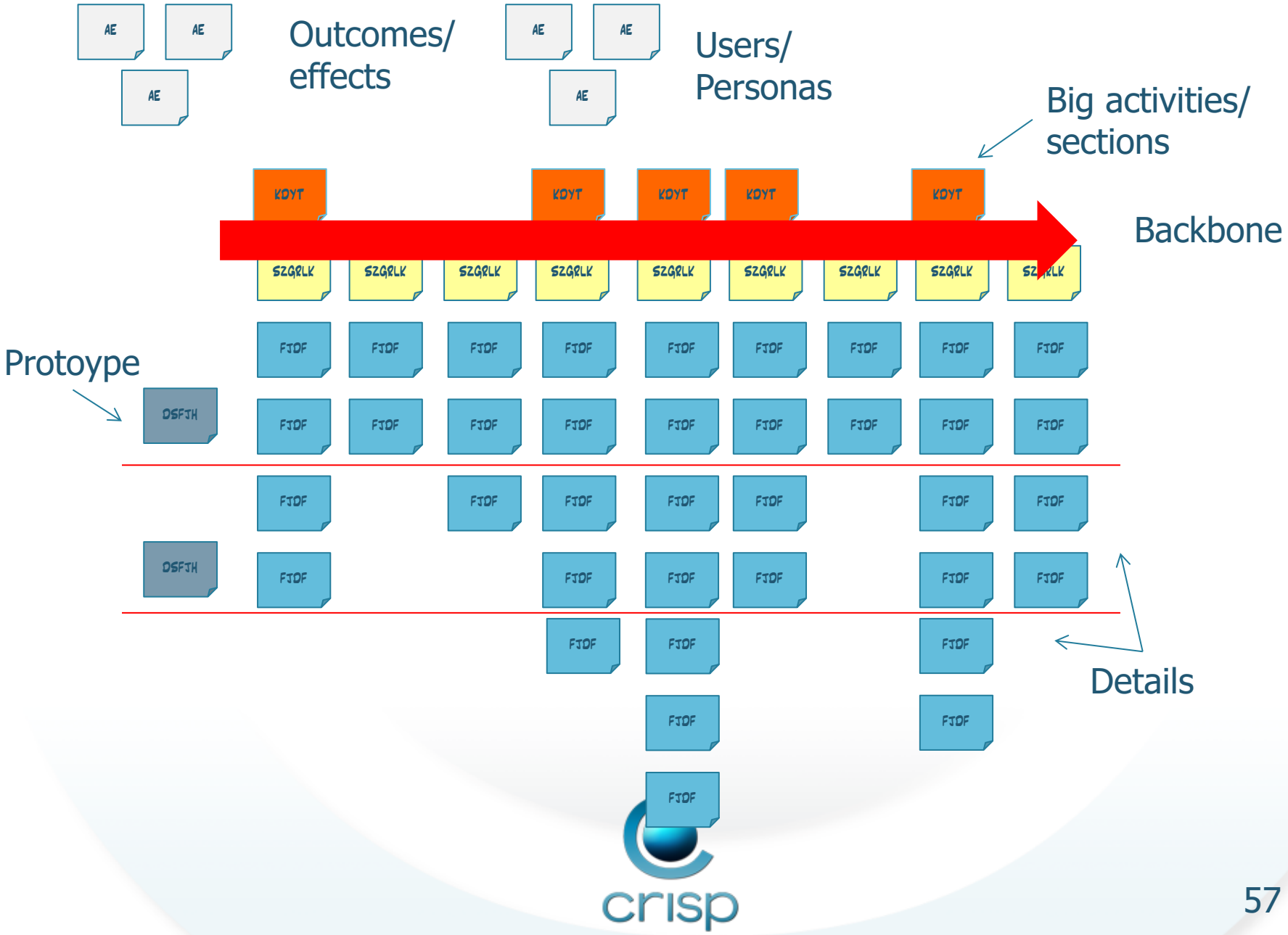


crisp

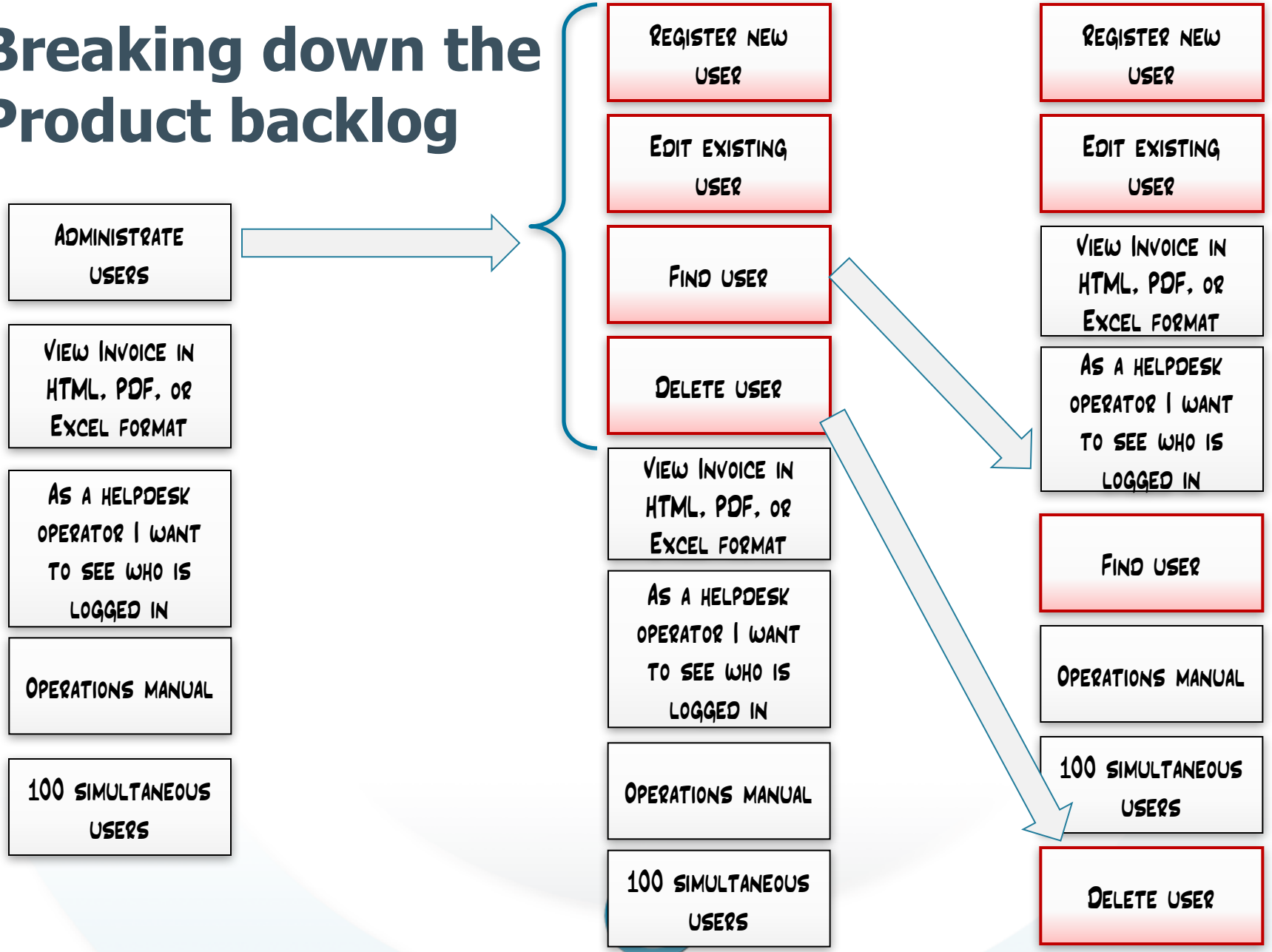
User Story Mapping

- See the whole picture
- Width and depth
- Creates shared understanding



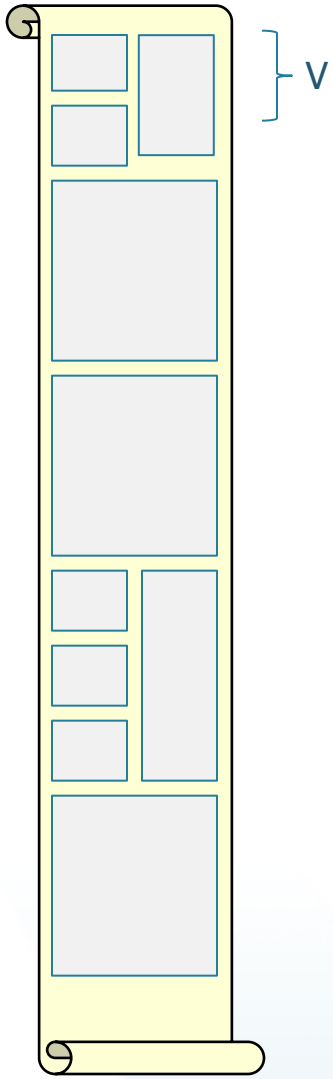

AE
AE
AE
Outcomes/
effects
AE
AE
AE
Users/
Personas
Big activities/
sections
KDYT
Backbone
SZGRLK
FJDF
Protoype
DSFJH
DSFJH
Details

Breaking down the Product backlog

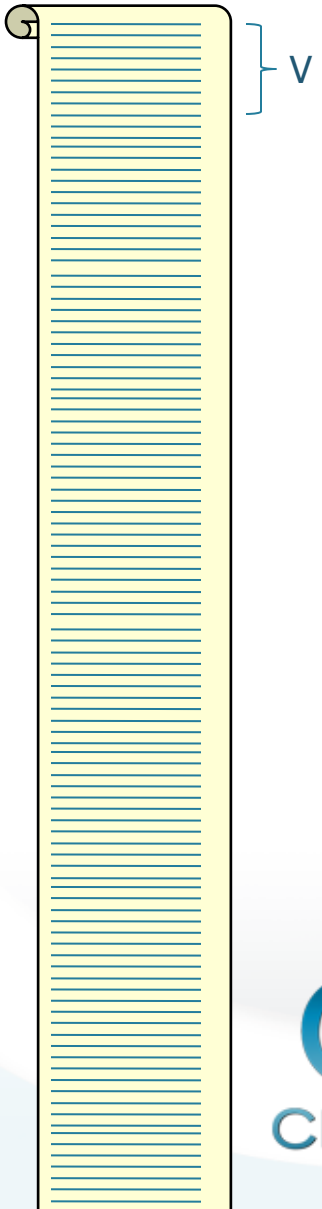


Balance the product backlog

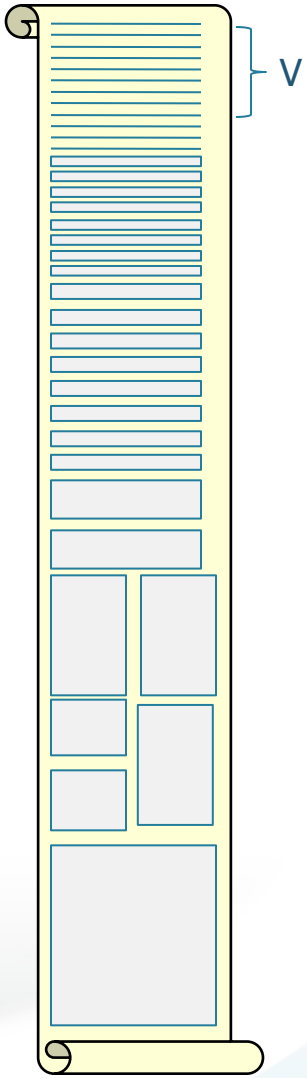
Too big items?



Too many items?

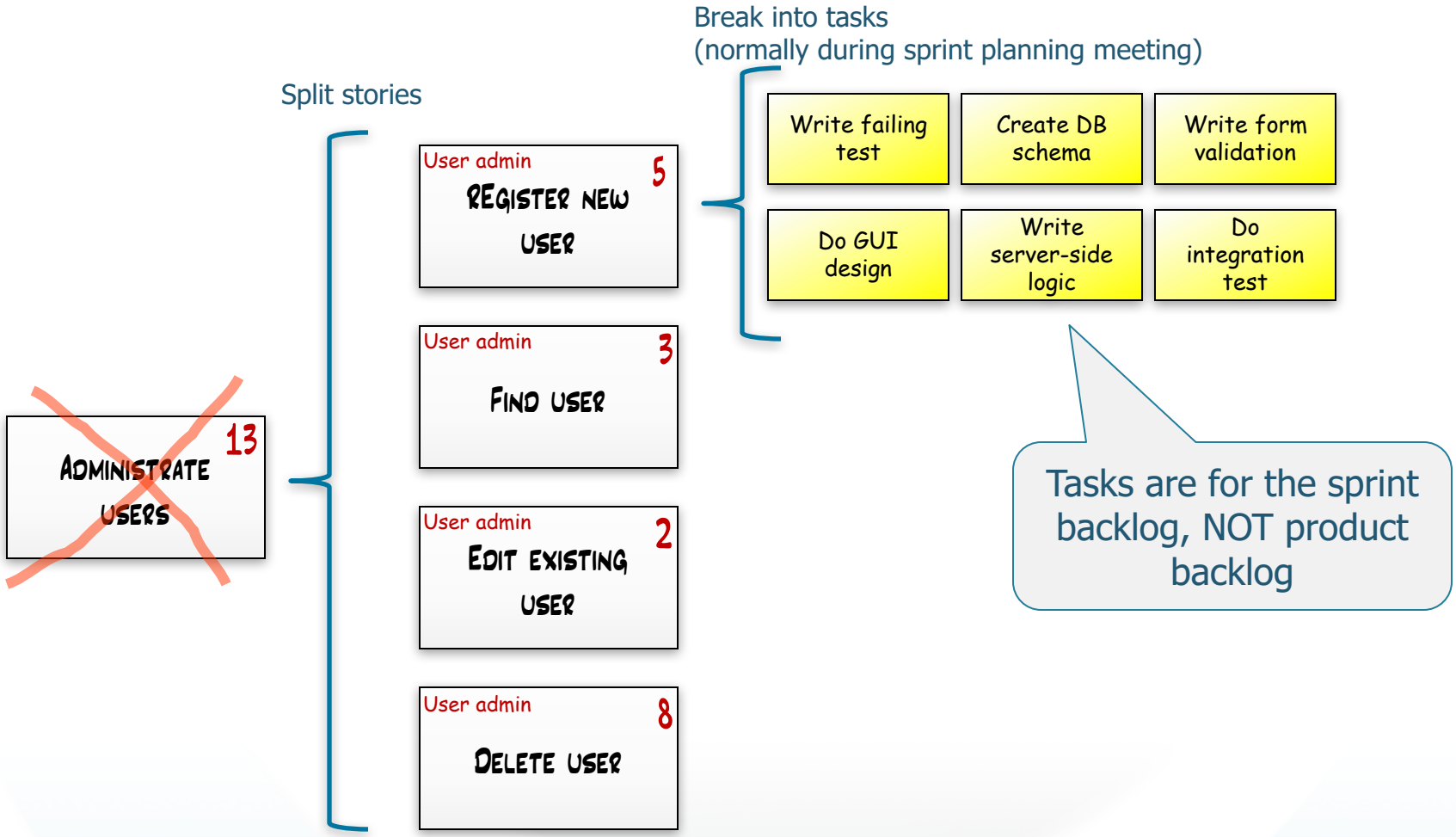


Balanced



crisp

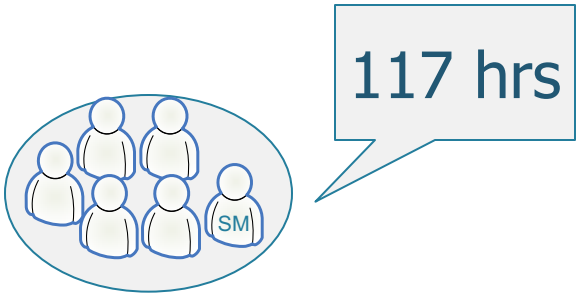
Splitting stories and breaking out tasks



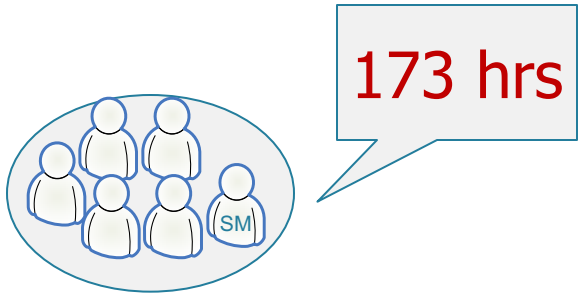
Estimation

Specification length

Spec



Same spec – more pages

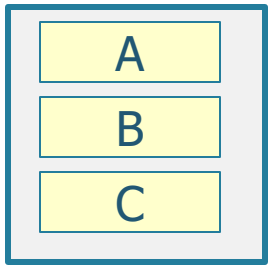


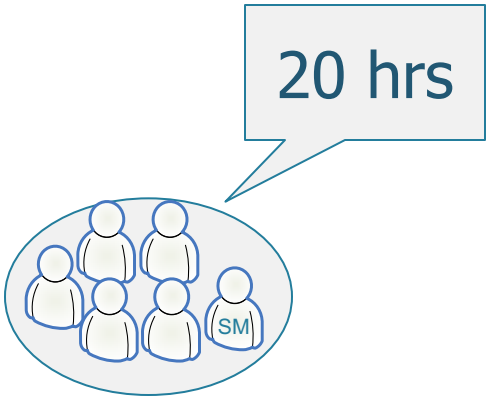
Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

crisp

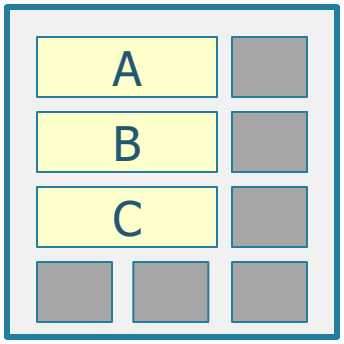
Irrelevant information

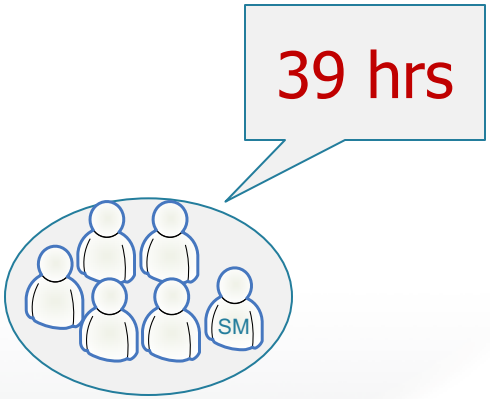
Spec 1





Same spec
+ irrelevant details



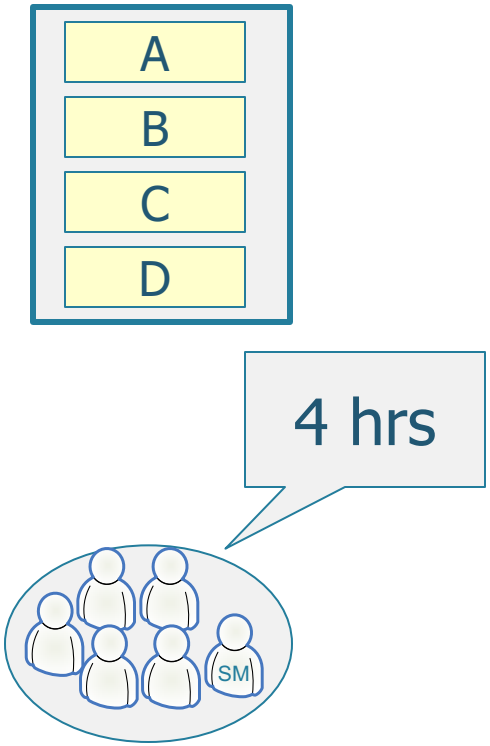


Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

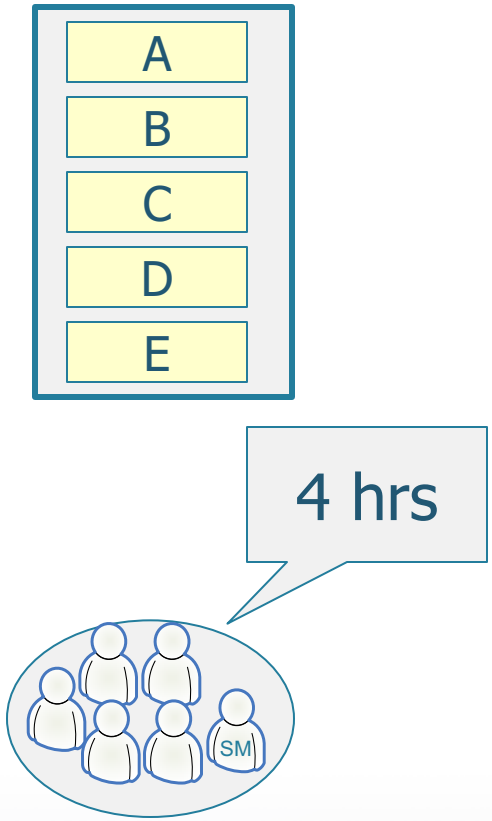
crisp

Extra requirements

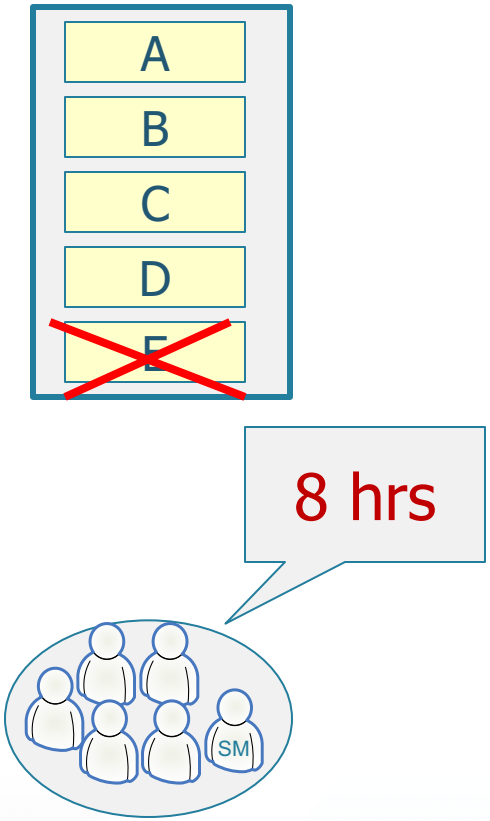
Spec 1



Spec 2

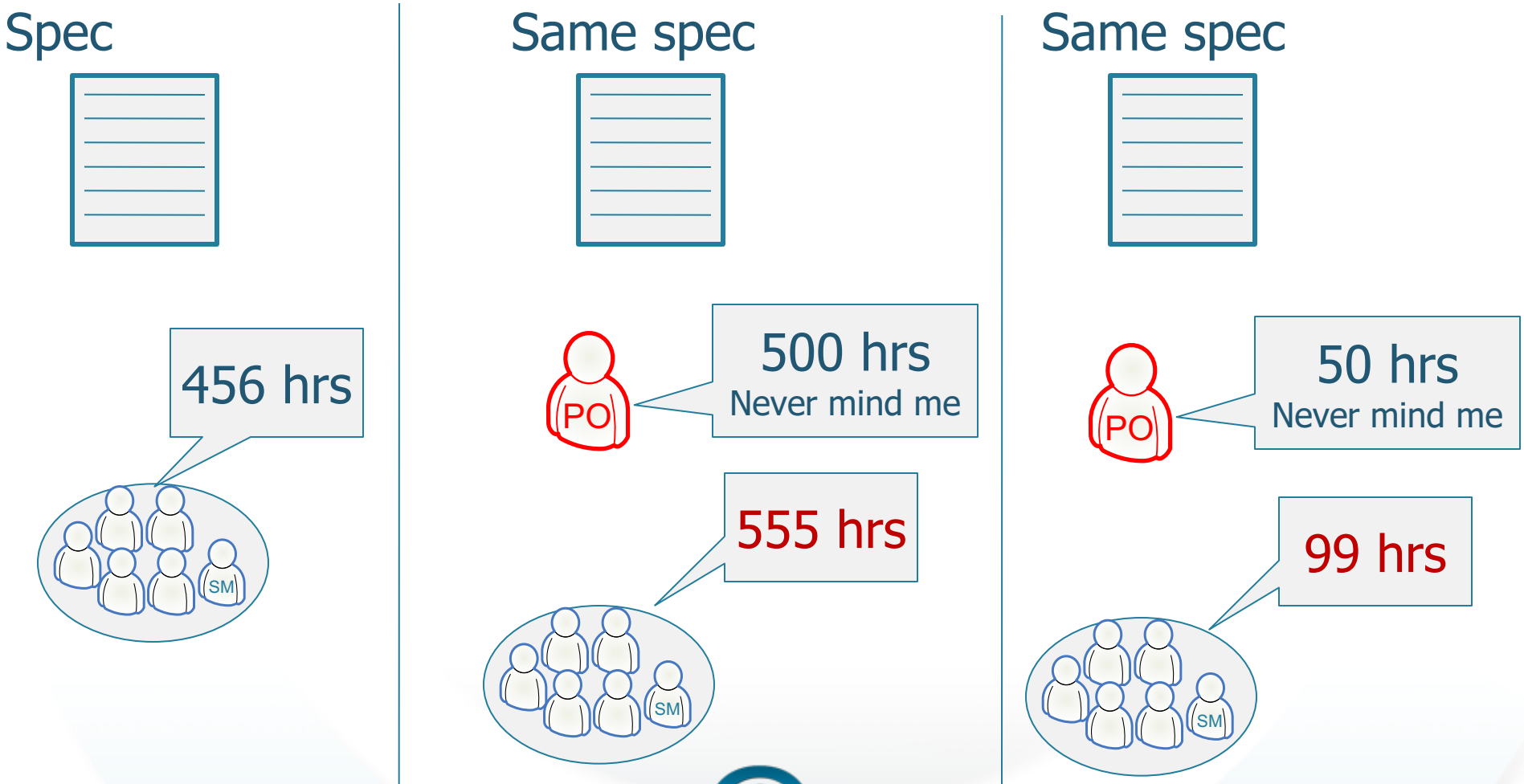


Spec 3



Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

Anchoring



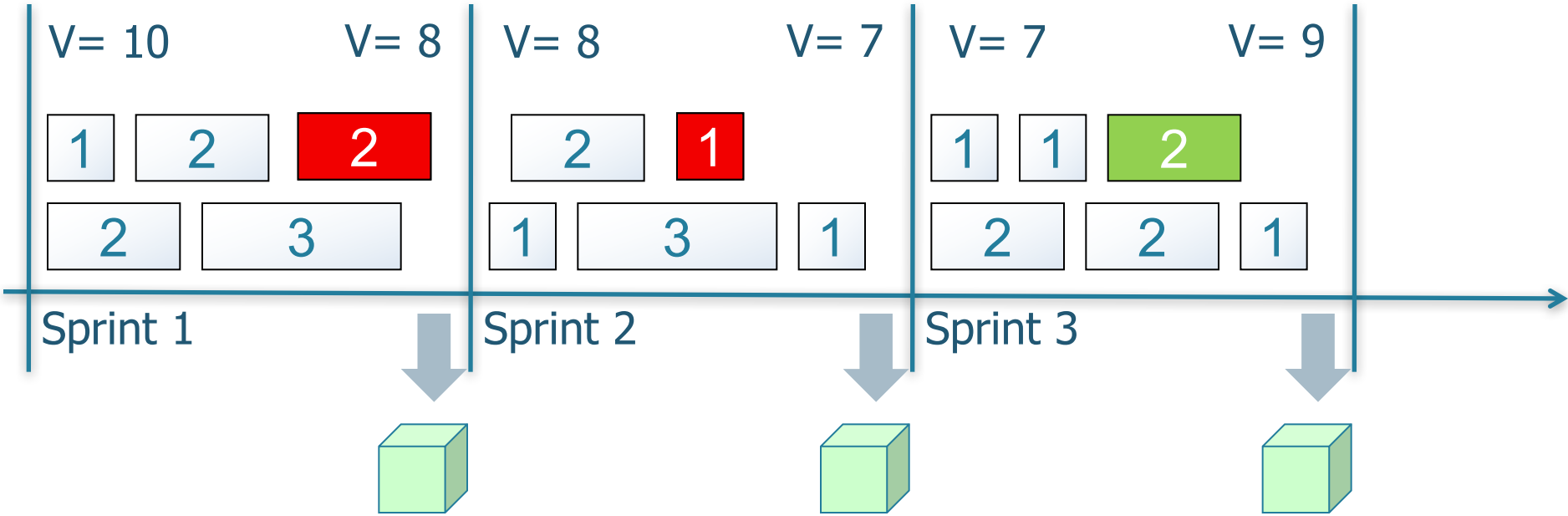
Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

Agile estimating strategy

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

http://planningpoker.crisp.se

Velocity



Likely future velocity:
7-9 per sprint

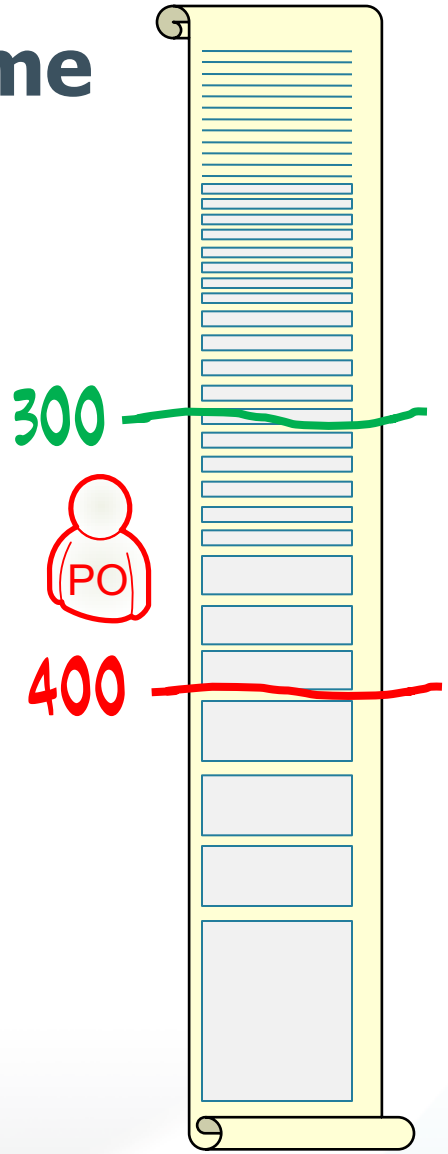
Planning

Release planning – fixed time

- Today is Aug 6
- Sprint length = 2 weeks
- Velocity = 30 - 40

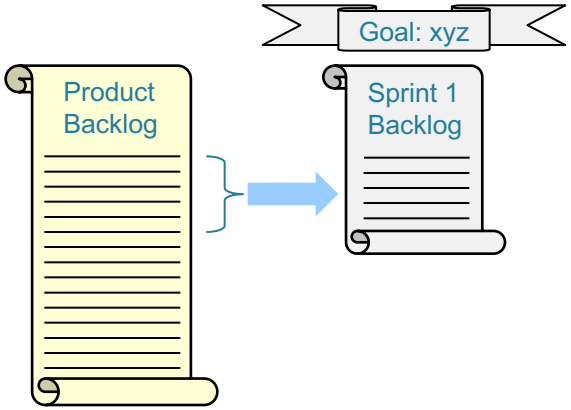
What will be done by X-mas?

(10 sprints)



crisp

Sprint planning meeting



Jackass team, sprint 15

Sprint goal
- Beta-ready release!

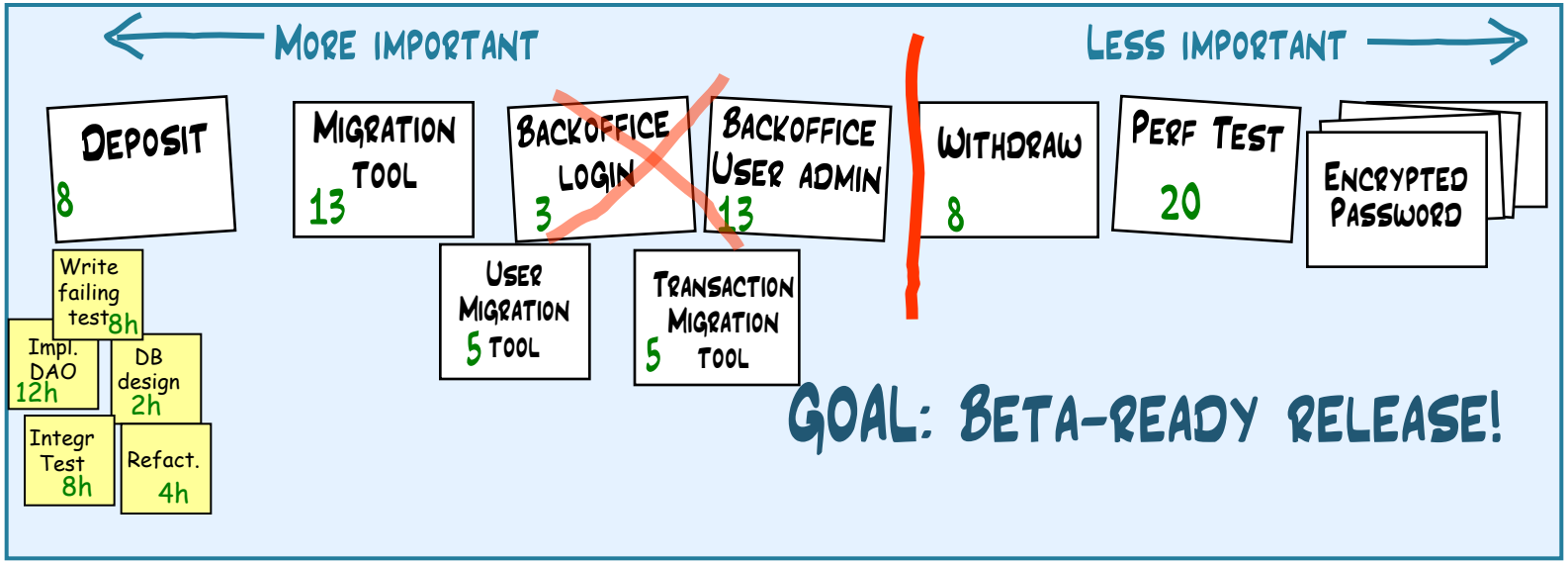
Sprint backlog
- Deposit (5)
- Migration tool (13)
- Backoffice login (3)
- Backoffice user admin (5)

(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

Sprint planning meeting - example



- Goal
- Present backlog
- Reprioritize, Re-estimate, split stories, combine stories
- Break out tasks
- Estimate velocity, draw the line

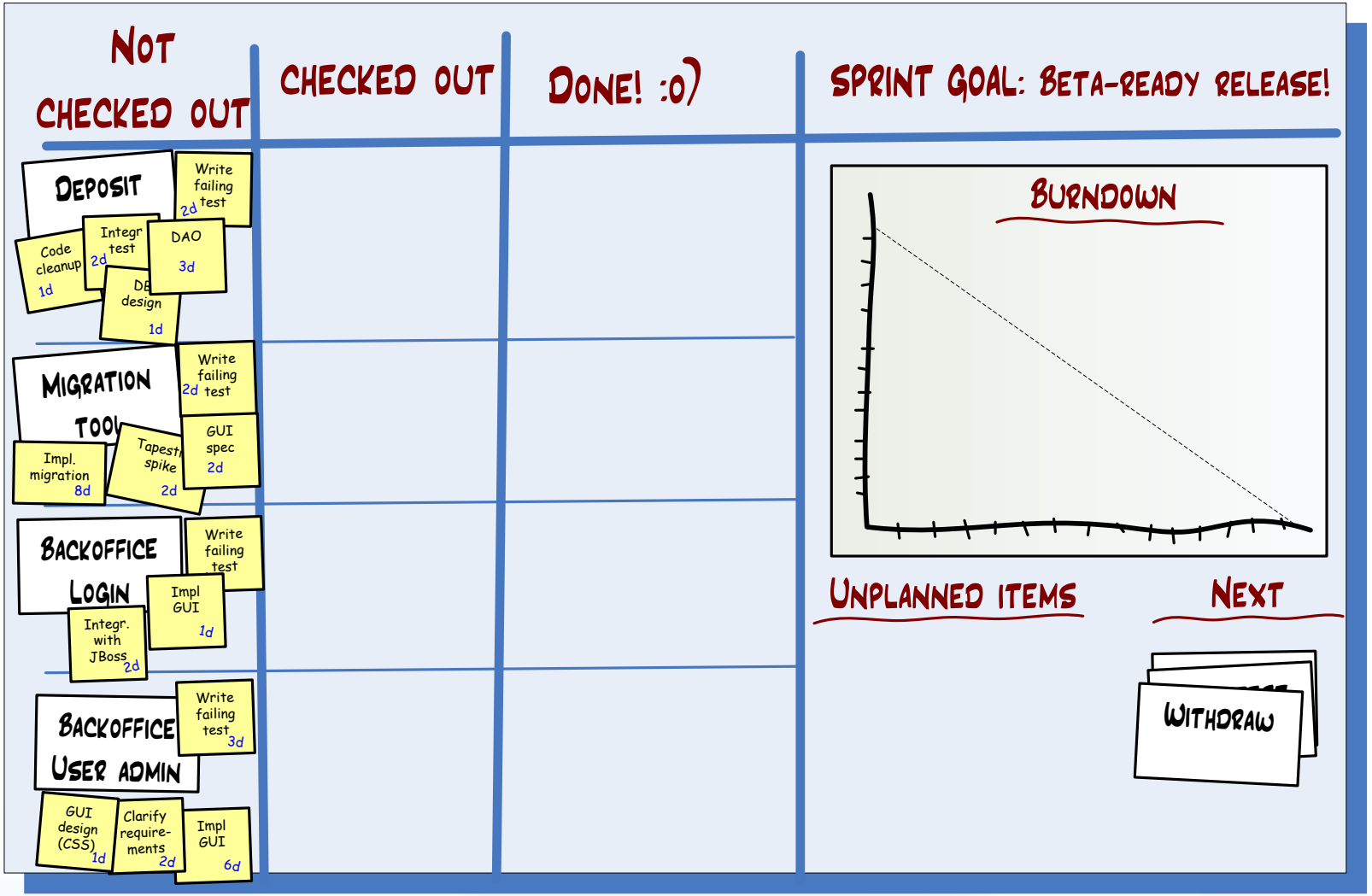
Not like this

Shared Understanding and collaboration at Atlassian (Agile planning tool company)

crisp

Sprint Backlog & Daily Scrum

Sprint backlog – day 0



crisp

Daily Scrum meeting

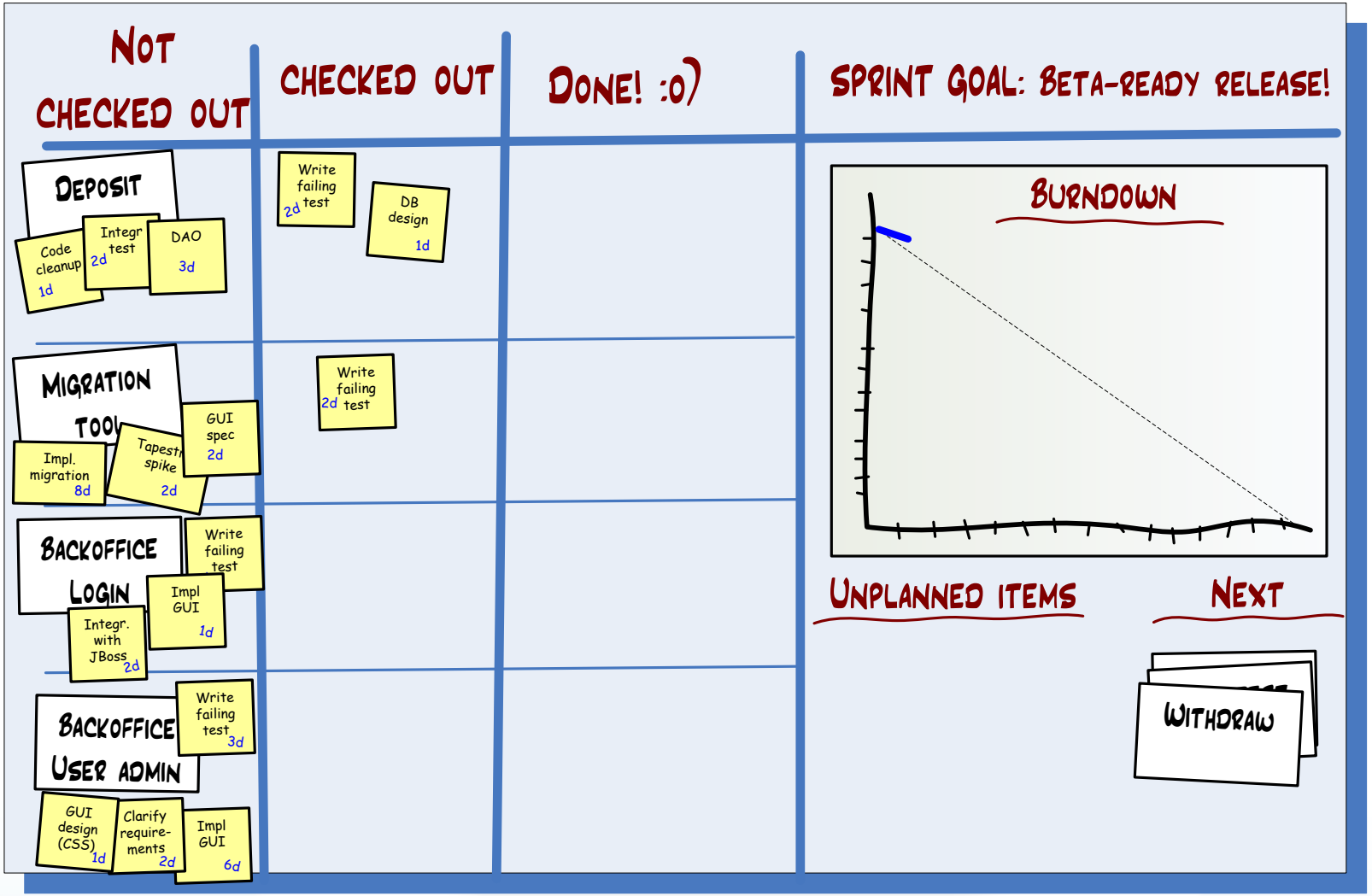
15 minutes

- What did I accomplish yesterday?
- What will I accomplish today?
- What's stopping me?

crisp

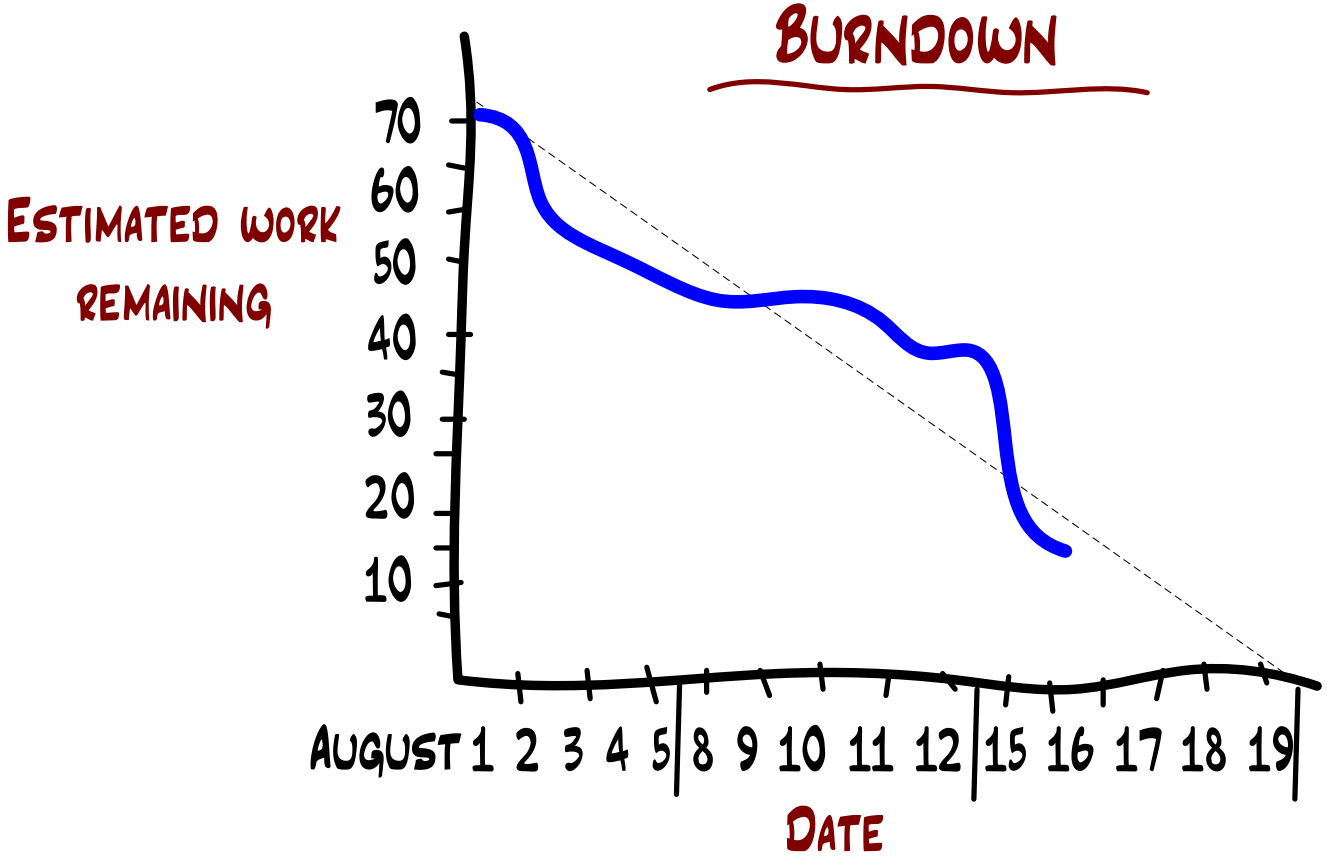
Work area layout

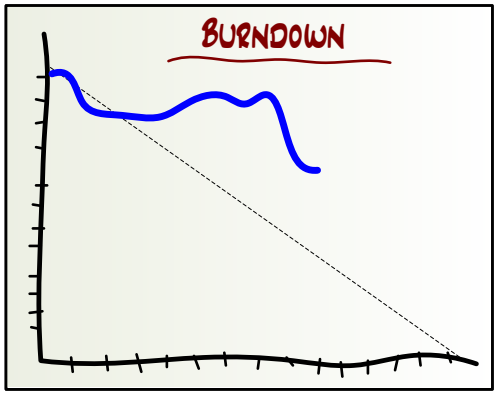
Sprint backlog – after 1st meeting

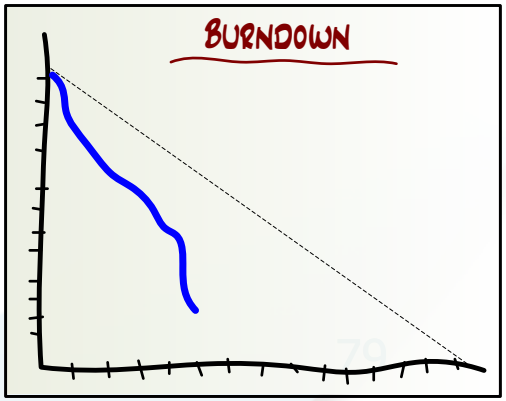


crisp

Sprint burndown chart

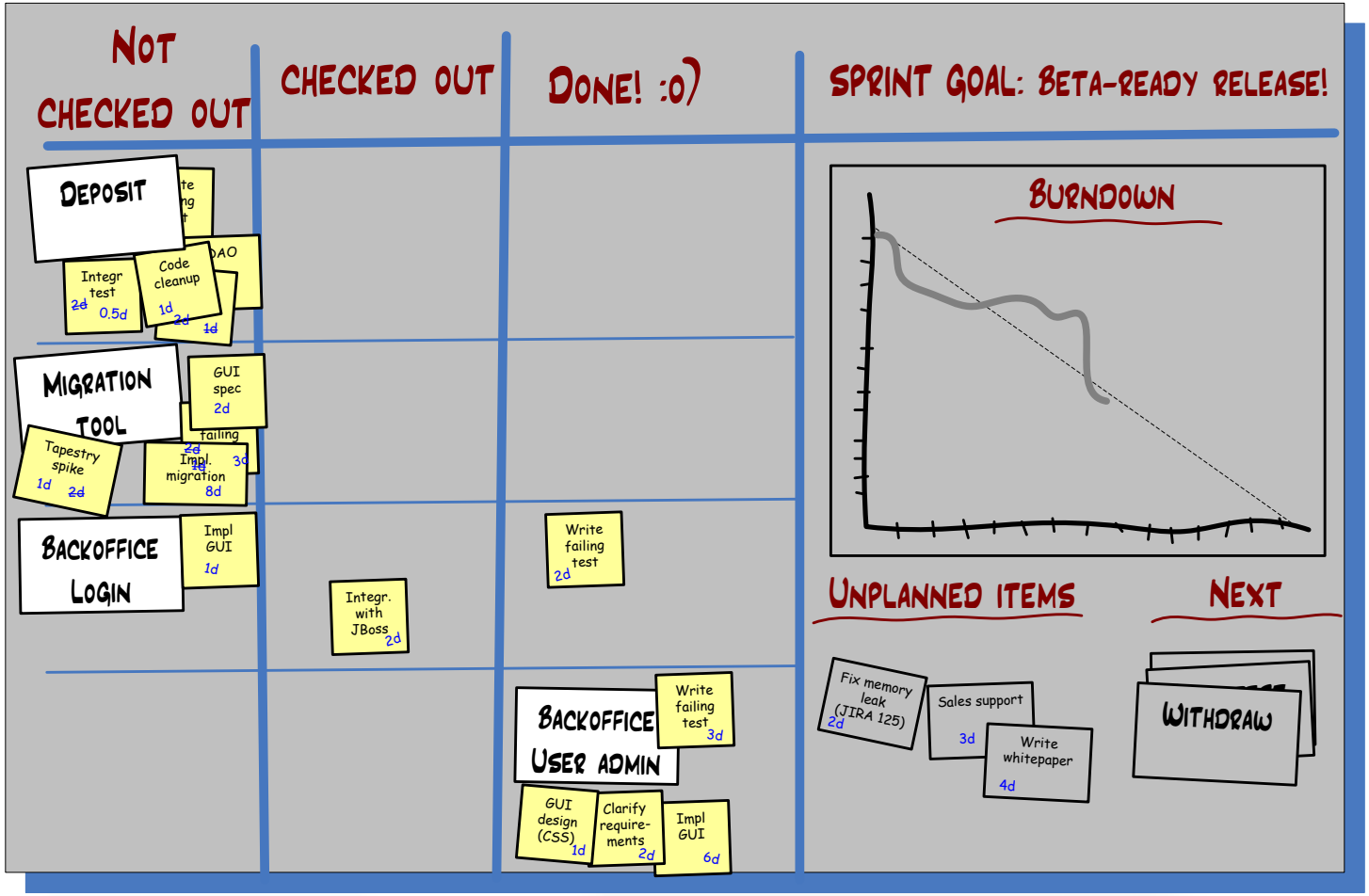




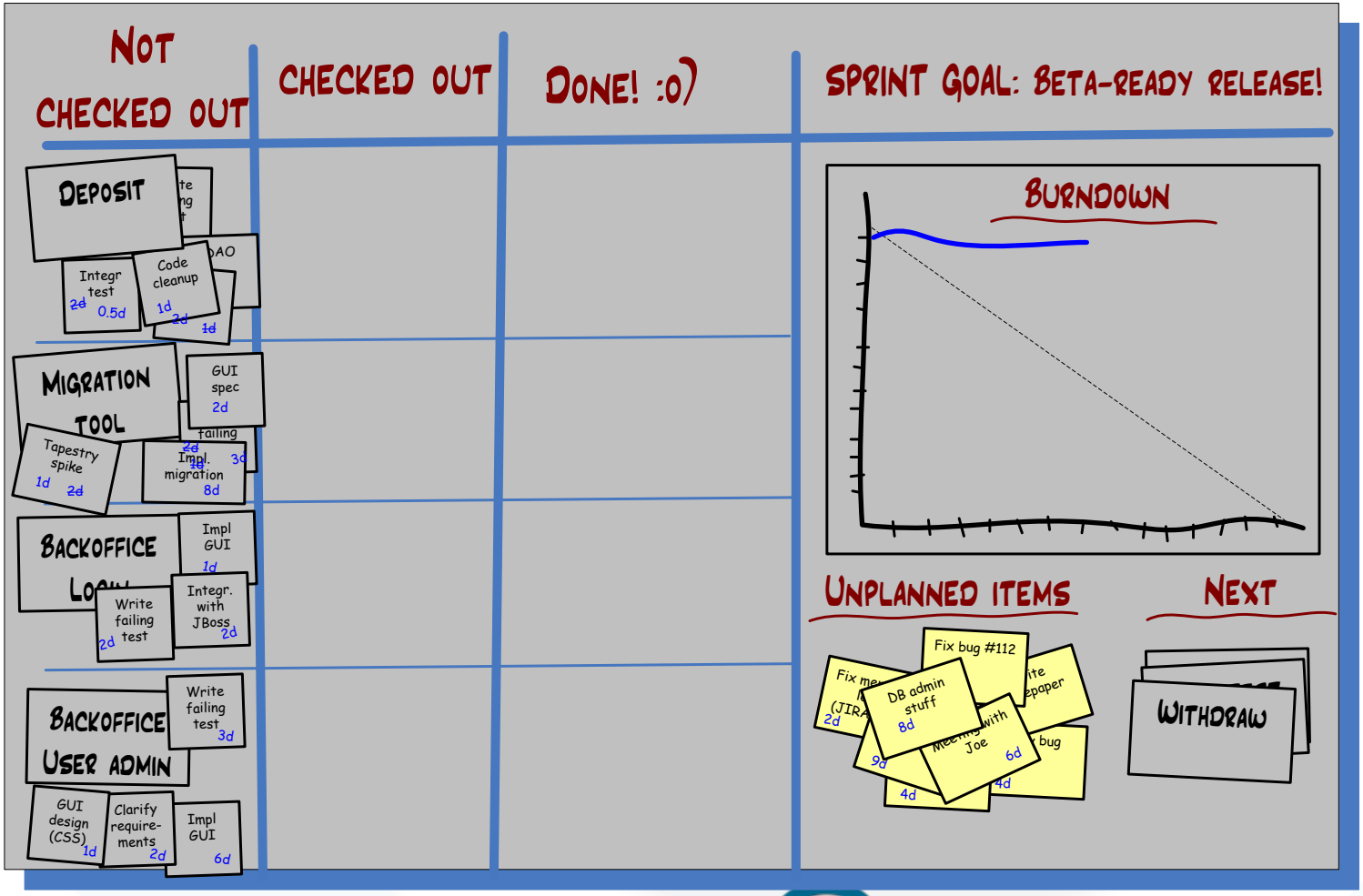


crisp

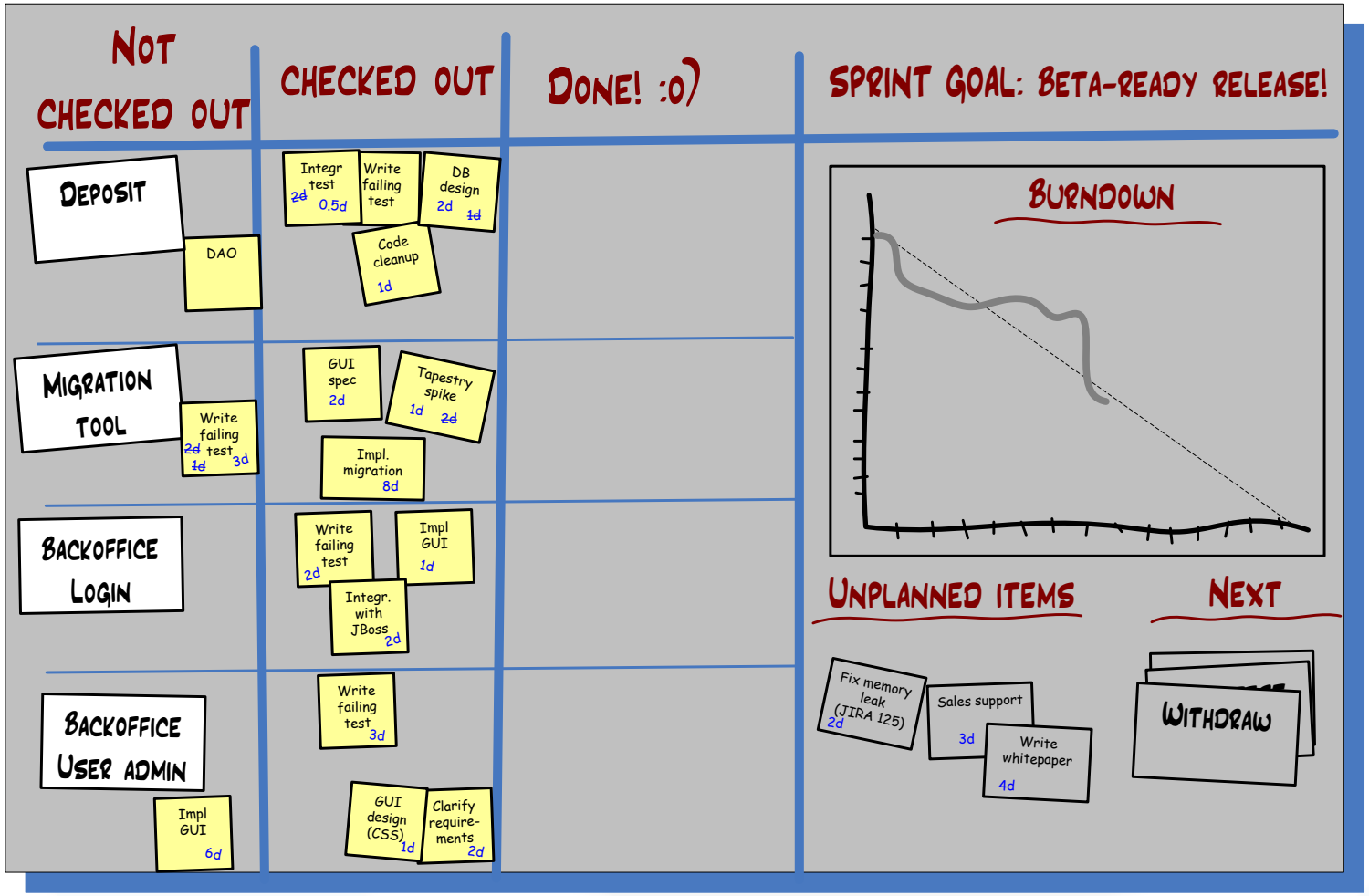
Warning sign #1



Warning sign #2



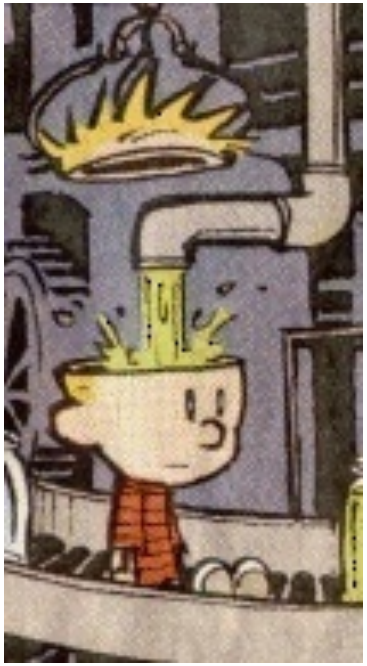
Warning sign #3



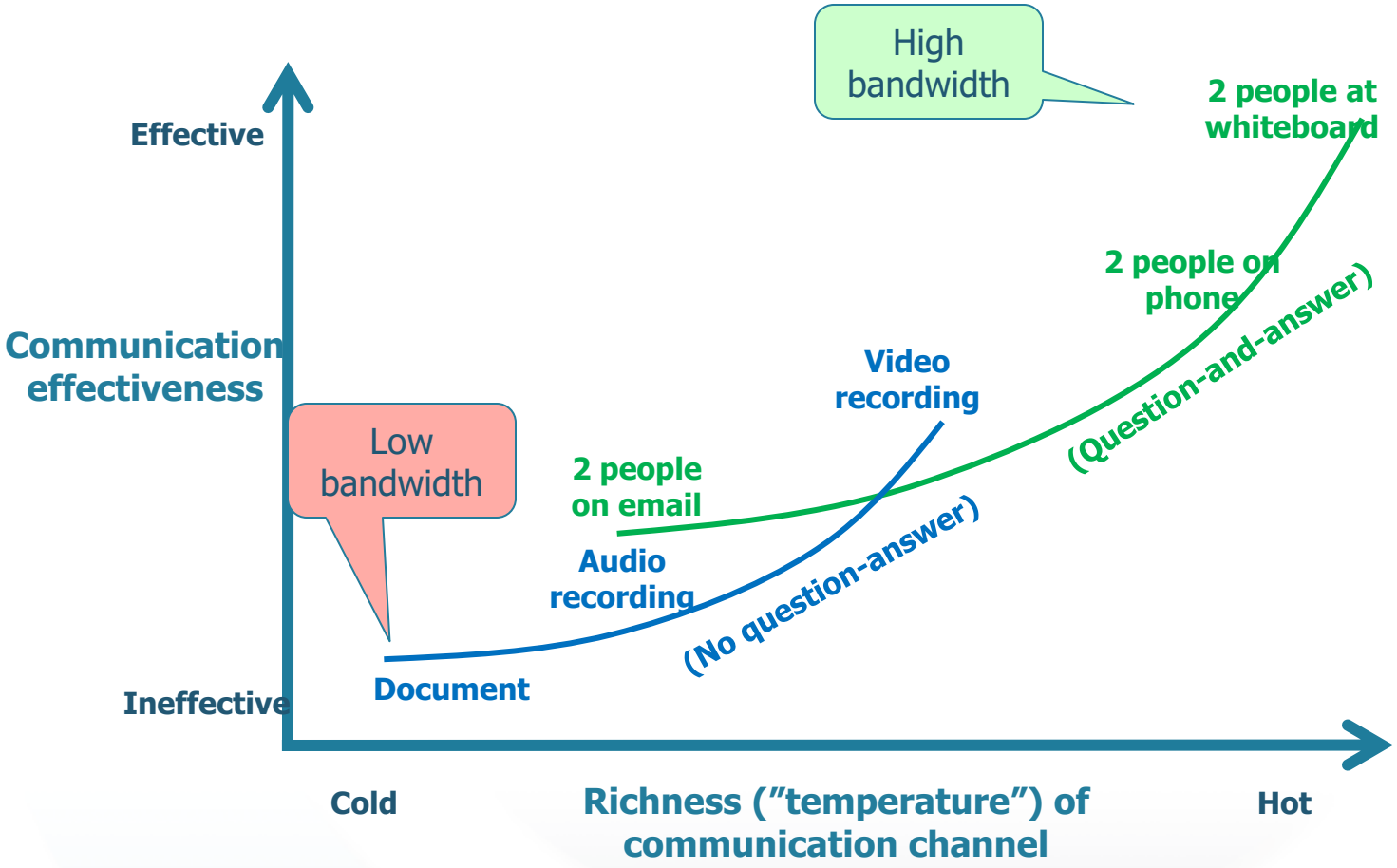
WAIT A SEC

How is that burndown calculated?

That's it!



Communication effectiveness



Source: research from McCarthy and Monk (1994)