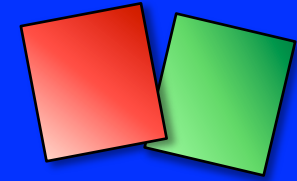


Scrum & XP

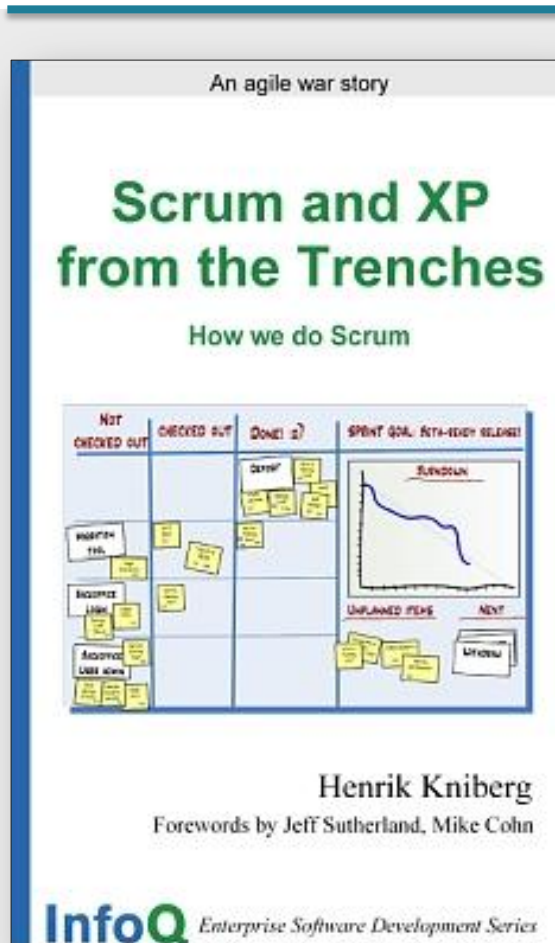
Beyond the trenches

JFokus, Stockholm
2011-02-15

PREPARATION



Make sure you have a
Pink & Green stickynote

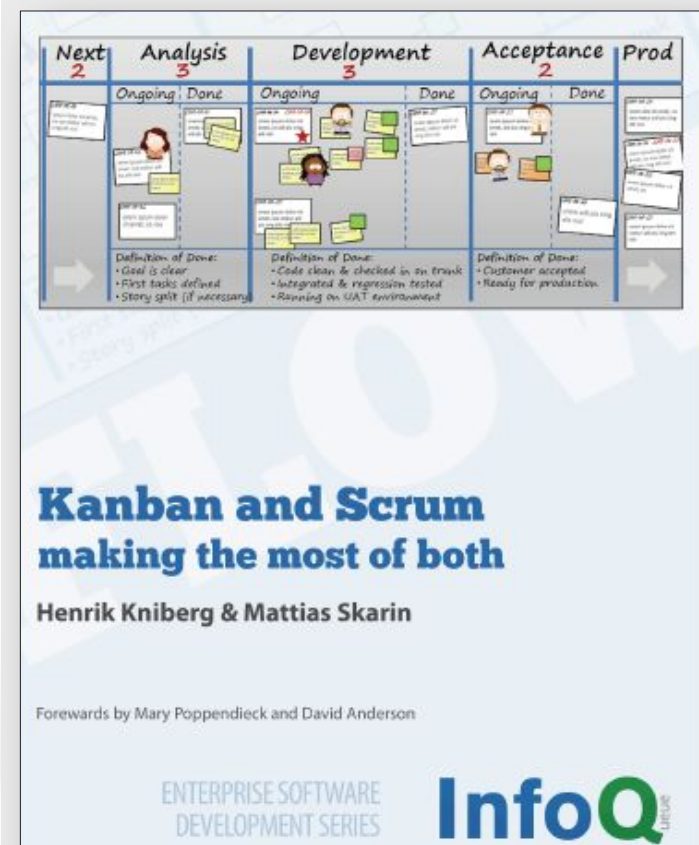


Henrik Kniberg
Agile/Lean coach
www.crisp.se

Agile Board of
Alliance® directors



henrik.kniberg@crisp.se
070 4925284





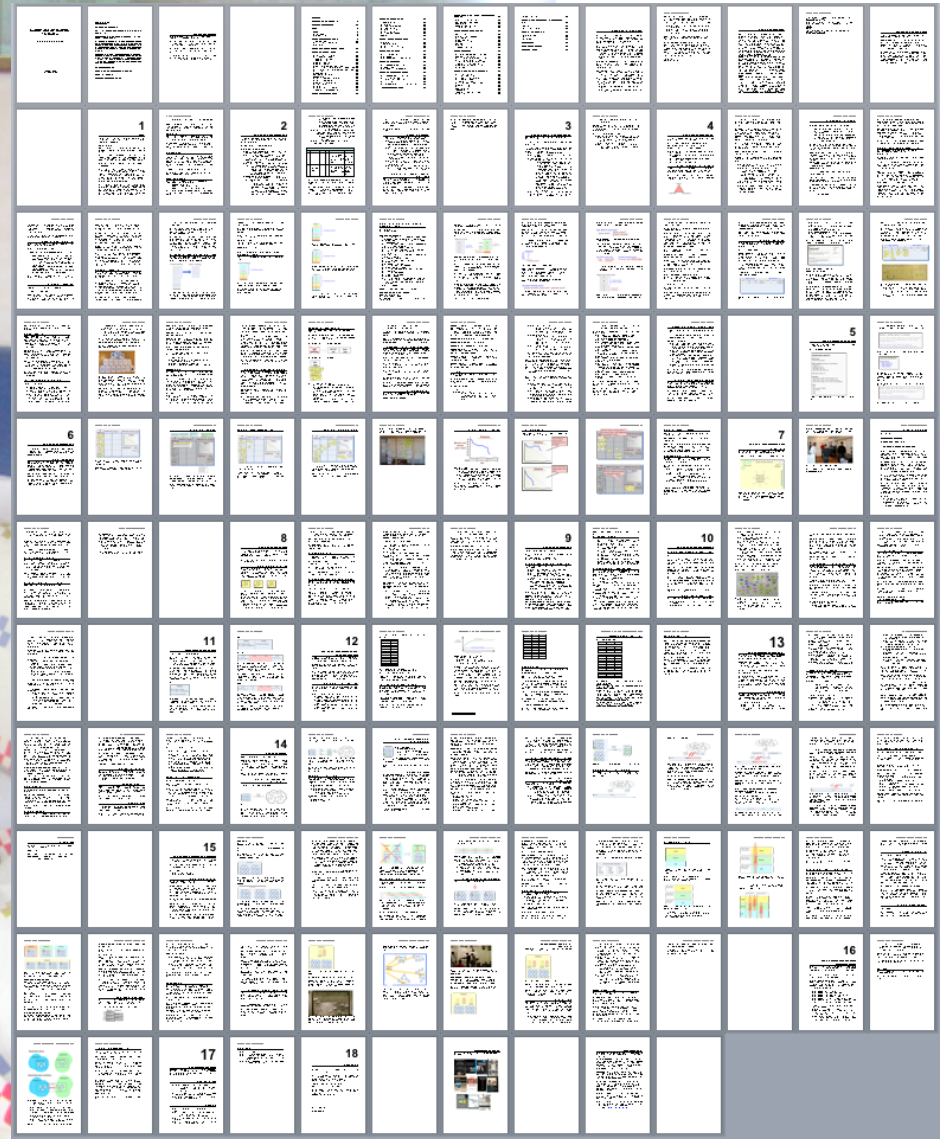
Henrik Kniberg



Picture from <http://www.geneticanomaly.com/RPG-Motivational/slides/fearlesspenguin.html>

Once upon a weekend

Nov 10-12, 2006



Henrik Kniberg

crisp

5



Scrum and XP from the Trenches

how we do Scrum



Henrik Kniberg
henrik.kniberg@crisp.se
www.crisp.se

Date: Monday, Nov 13, 2006

from: Henrik

to: scrumdevelopment@yahoogroups.com

subject: Scrum & XP from the trenches - how we do Scrum

I've written a paper (well more like a small book) describing lessons learnt after a year of Scrum experimentation with a group of 40 developers. Includes details on how we approached multi-team sprint planning, testing, retrospectives, etc.

Here's the final draft:

<http://www.crisp.se/henrik.kniberg/ScrumAndXpFromTheTrenches.pdf>

Any feedback is welcome!

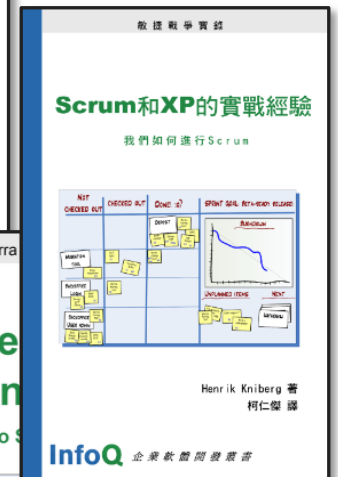
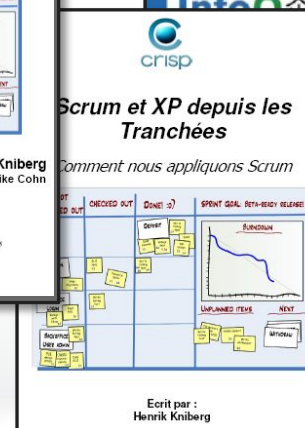
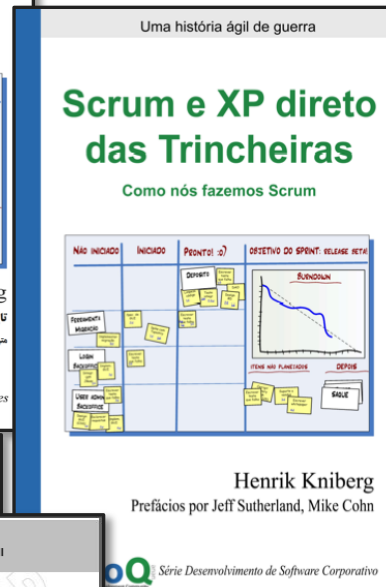
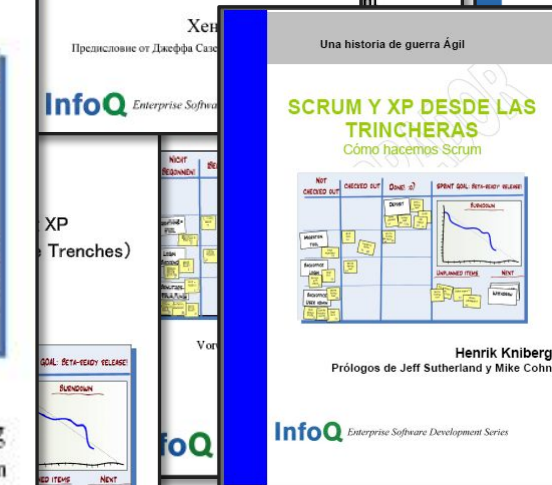
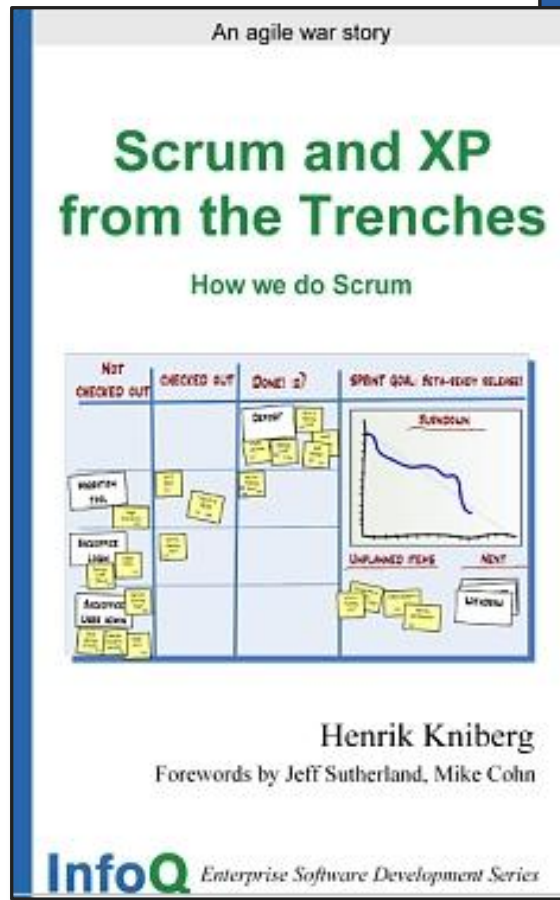
Those of you who are authors, any ideas on what I should do with a paper like this? Haven't done this kind of stuff before...

/Henrik

4 years later...

250,000 downloads

12 languages



Henrik Kniberg





Discovery #1

Other people were discovering the same solutions, independently



When I read your book I recognized all the problems and solutions, we went through the same journey!

Discovery #2

Most people don't read disclaimers

Disclaimer

This document does not claim to represent “the right” way to do Scrum! It only represents one way to do Scrum, the result of constant refinement over a year's time. You might even decide that we've got it all wrong.

“this is the paper to read if you want to avoid learning Agile by trial and error”



“We've have adopted it as our tech bible here at my company”



Discovery #3

Copy/paste worked better than I had thought!



"I've started using Scrum in my company and it's been a success thanks to your valuable information"

But not perfectly...



We spend a lot of time arguing over Focus Factor

Who cares about Focus Factor? Why do you waste time on that?

We learned about it from your book...

Oops...

Well, we stopped doing that right after the book came out



Purpose of this presentation

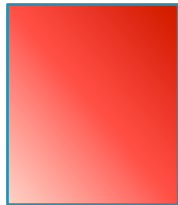
- Many of the solutions in the book turned out to be pretty much universally applicable
- However, I've learned a lot since 2006 and there some things that I would have done differently if I could go back in time.
- Purpose of this presentation is to share these insights with you.

Let's compare notes.

When you see this signal
Vote!



= I agree with this recommendation.



= I disagree with this recommendation

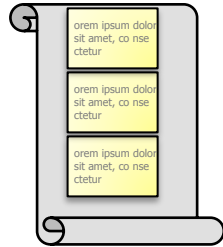
Recommendation #1:

Limit all queues

12

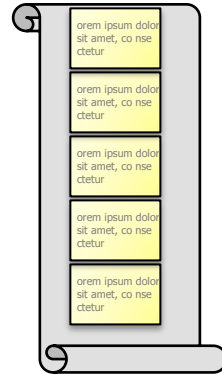
3 items

6 orderings



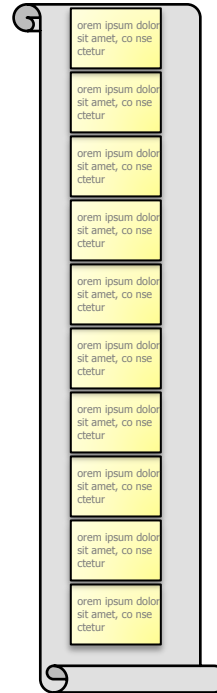
5 items

120 orderings



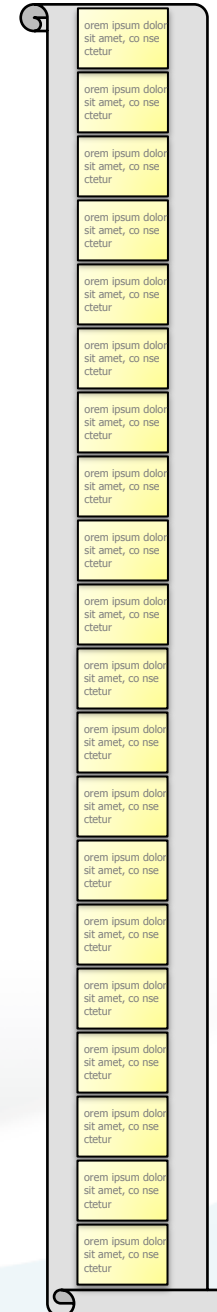
10 items

3,628,800 orderings



20 items

2,432,902,010,000,000,000 orderings

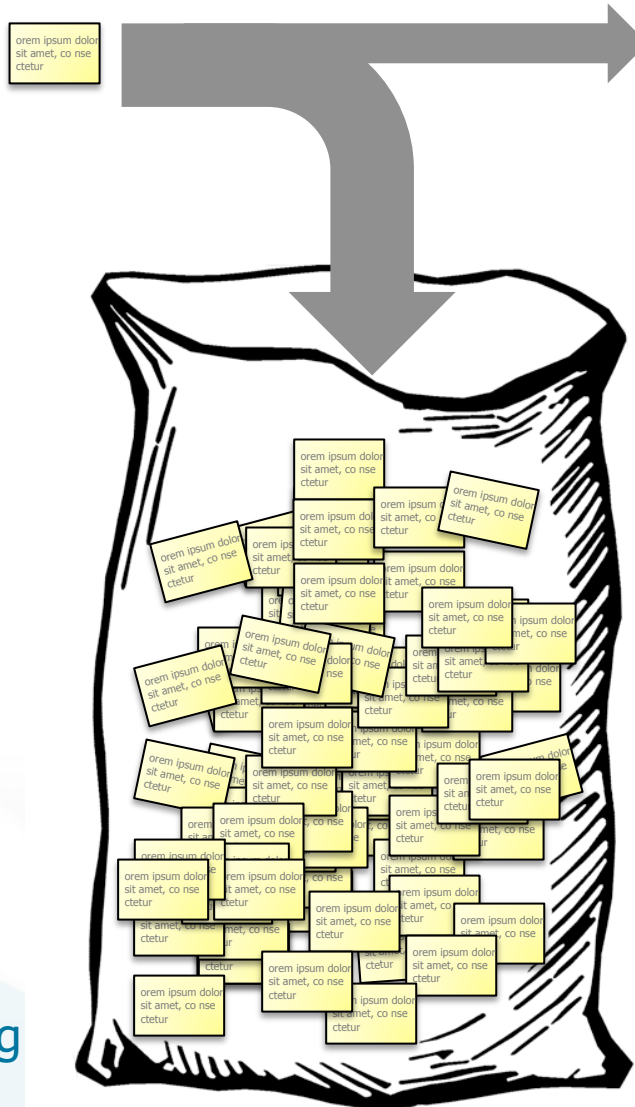
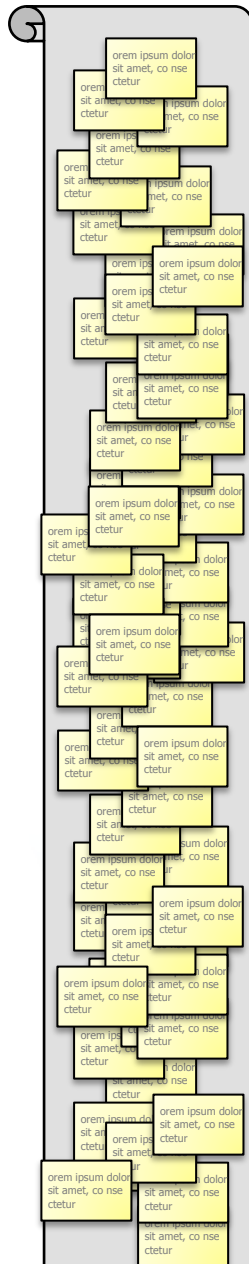


Product backlog or bug database

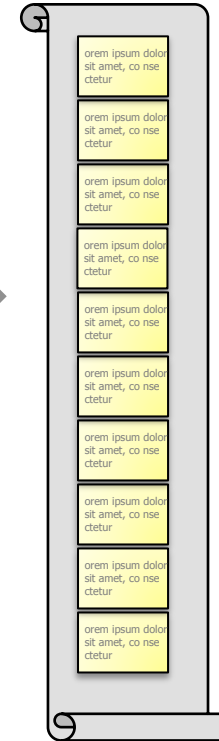
Length 100+

9×10^{157} orderings

(number of atoms in the universe: 4×10^{80})



Top 10



Recommendation:
Limit all queues



Henrik Kniberg

Recommendation #2:

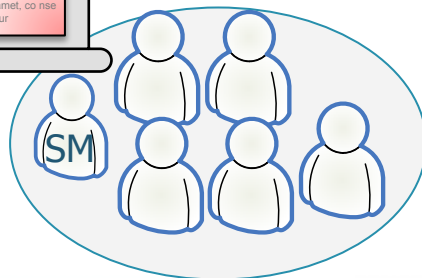
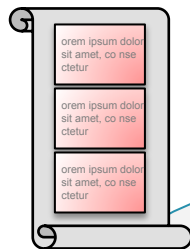
**Visualize top 3
impediments**

15

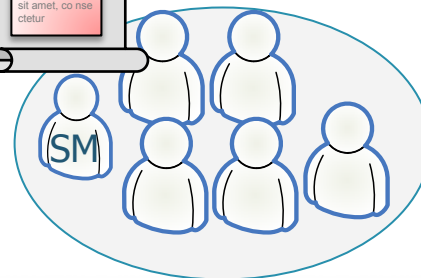
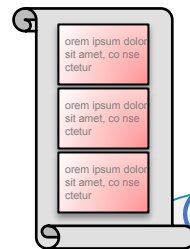
Manager



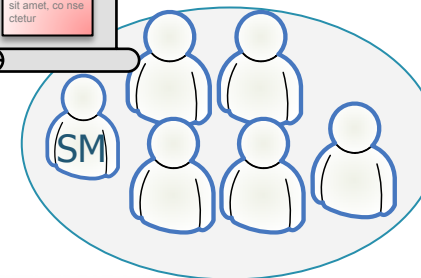
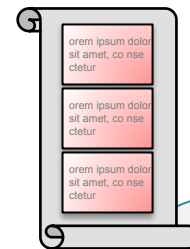
Top 3
impediments

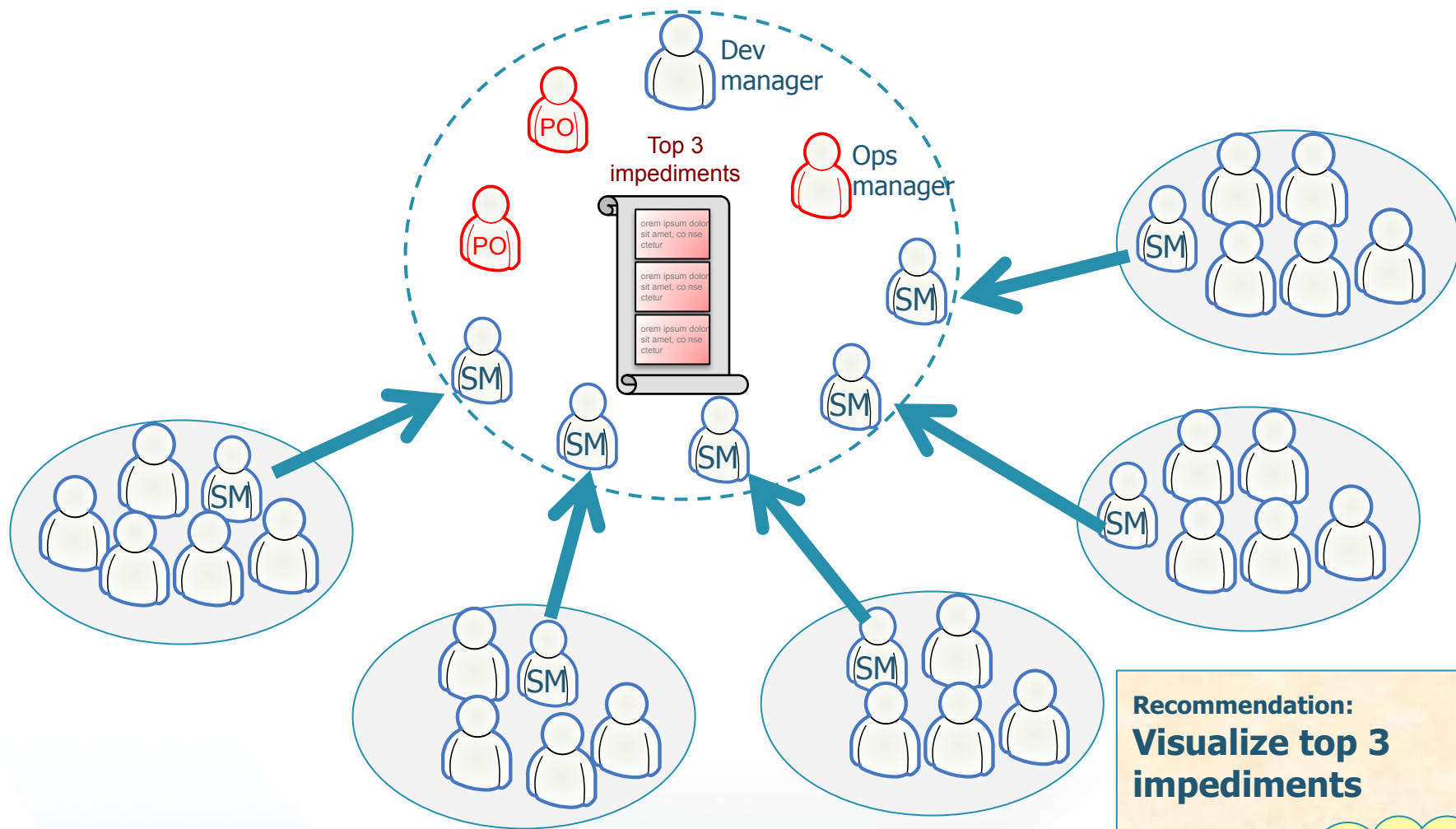


Top 3
impediments



Top 3
impediments





Recommendation:
Visualize top 3 impediments

Agree?

No! **Yes!**

Recommendation #3:

**Continuously
measure employee
satisfaction**

18

Hur känns Crisp för dig just nu? Vad säger magkänslan?						
Detta är en stående poll. Uppdateras regelbundet. Snittvärdet loggas regelbundet under Historikflik.						
Vem	Nöjd/nivå (1-5)	Senast uppdaterat (Invaligt)	Vad känns bäst just nu?	Vad känns sämst just nu? (Invaligt)	Vad skulle öka din nöjdhet? (Invaligt)	Övriga kommentarer (Invaligt)
XXX	4	23-mar	Postkod i Sthlm seglat upp som huvudkund. Bra jobb Mats P1	April, Maj är så ockuperat att jag inte hinner följa Crisp.		
XXX	4	24-mar	Nära till jobbet, bra team. Lysande PO. Får chansen att jobba med spännande teknik (Terracotta)	Inte så sexigt förvaltningsuppdrag	Kortare arbetsresa :)	
XXX	4	9-apr	Har uppdrag. Inte så många på bänken längre.		Lägre avgifter	
XXX	3	26-jan	Nytt uppdrag!	Har varit utan uppdrag tot 7 månader sista året		
XXX	4	11-apr	Tivs bra på YYY. Kul med så många Crispare. Verkar som om det är i konsult djungeln.	Orolig för byte av säljare och hur det kommer att påverka försäljningen inför hösten	Vetskap om nytt uppdrag i höst.	Alla Crispare är så duktiga
XXX	5	24-mar	Konsulturer uppdrag än vi kan ta.	Hinner inte med allt jag skulle vilja göra	Dygnet skulle utökas till 48 timmar	

- Name (optional)
- Satisfaction level (1-5)
- Last updated (date)
- What feels best right now?
- What feels worst right now?
- What would improve your satisfaction level?



Jeff Sutherland

<http://scrum.jeffsutherland.com/2010/11/happiness-metric-wave-of-future.html>
<http://scrum.jeffsutherland.com/2010/12/scrum-inc-sprint-2-retrospective.html>

Happiness metric -
the Wave of the Future

Henrik Kniberg



Russel Ackoff

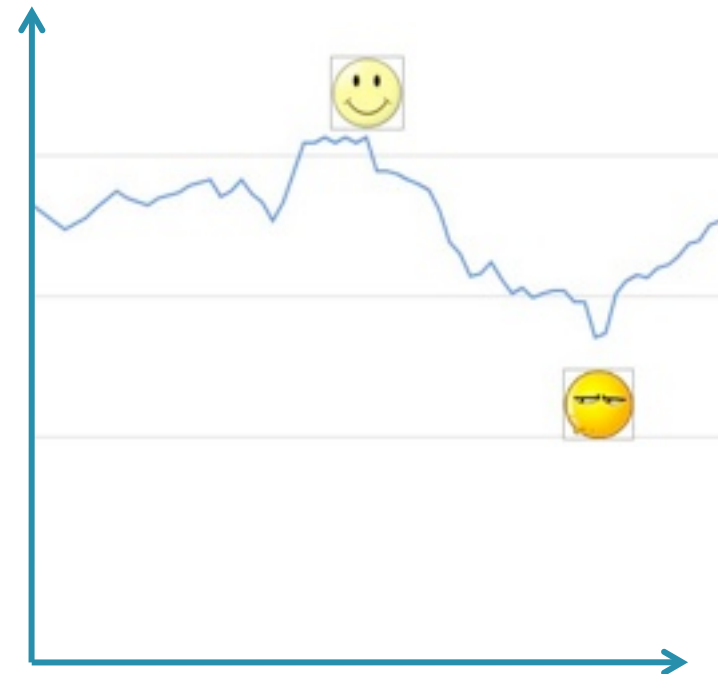
It is better to use imprecise
measures of what is wanted,
rather than precise measures of
what is not

Recommendation:
**Continuously
measure employee
satisfaction**

Agree?

No!

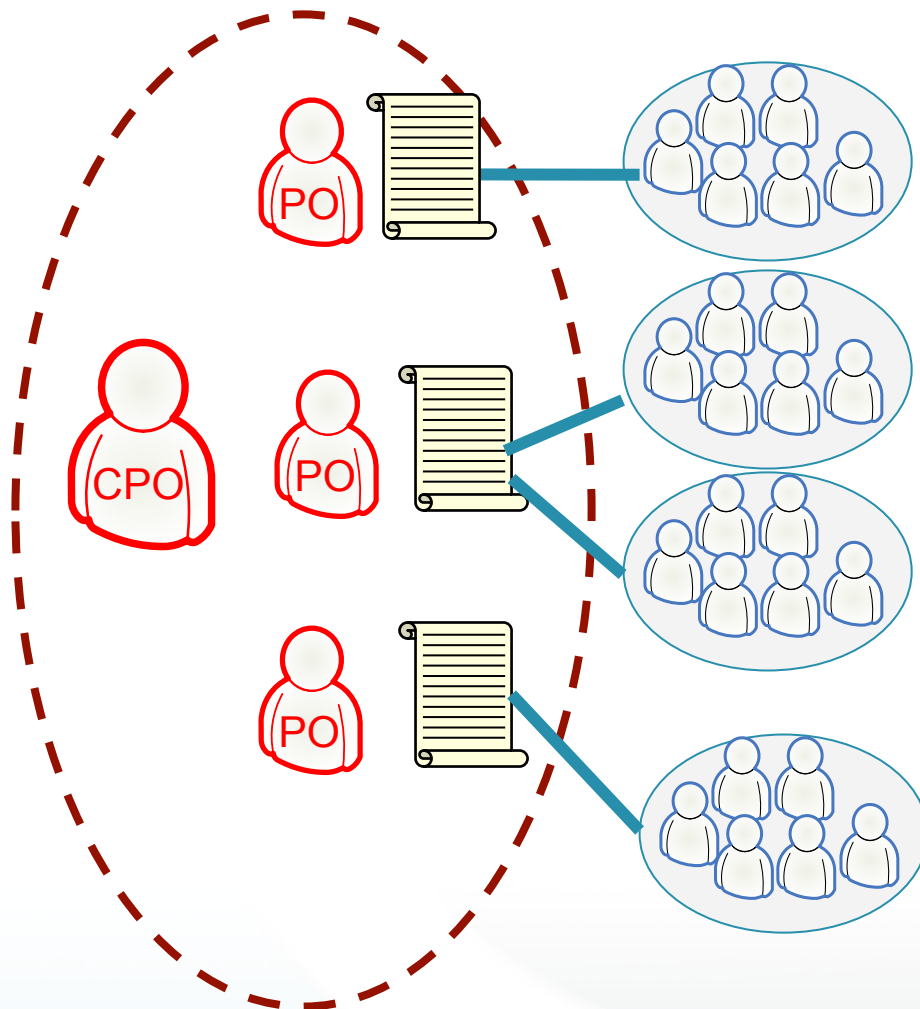
Yes!



Recommendation #4:

Have a chief product owner role

20



Recommendation:
Have a chief product owner role



Recommendation #5:

**Express backlog items as
User Stories**

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

8

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

Recommendation:

**Express backlog items
as User Stories**

Agree?

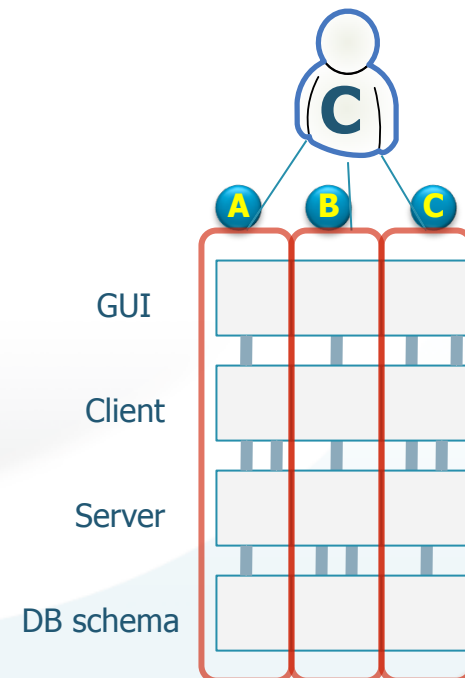
No!

Yes!



Independent
Negotiable
Valuable
Estimable
Small
Testable

Acronym courtesy of Bill Wake – www.xp123.com



Recommendation #6:

**Super stickynotes &
index cards & magic
chart & magnet tejp**

24



<http://www.clasohlson.se/Product/Product.aspx?id=136789378>



http://www.svanstroms.com/ecommerce/control/keywordsearch?SEARCH_STRING=2028901



http://www.svanstroms.com/ecommerce/control/keywordsearch?SEARCH_STRING=2631666&



http://www.svanstroms.com/ecommerce/control/keywordsearch?SEARCH_STRING=2400108

Recommendation:
**Super stickynotes &
 index cards & magic
 chart & magnet tape**

Agree?

No!

Yes!

25

Henrik Kniberg

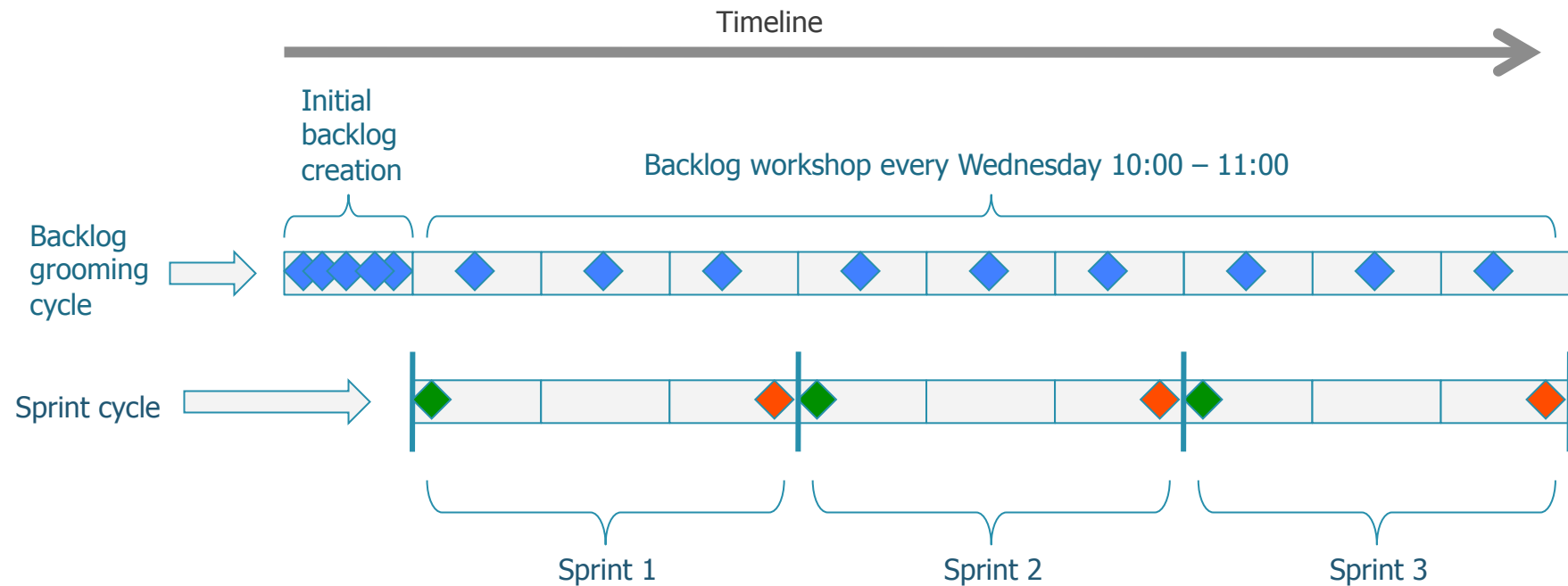


Recommendation #7:

**Estimate stories
before the sprint
planning meeting**

26

Backlog creation & grooming – sample schedule



Recommendation:
Estimate stories
before the sprint
planning meeting

Agree?

No!

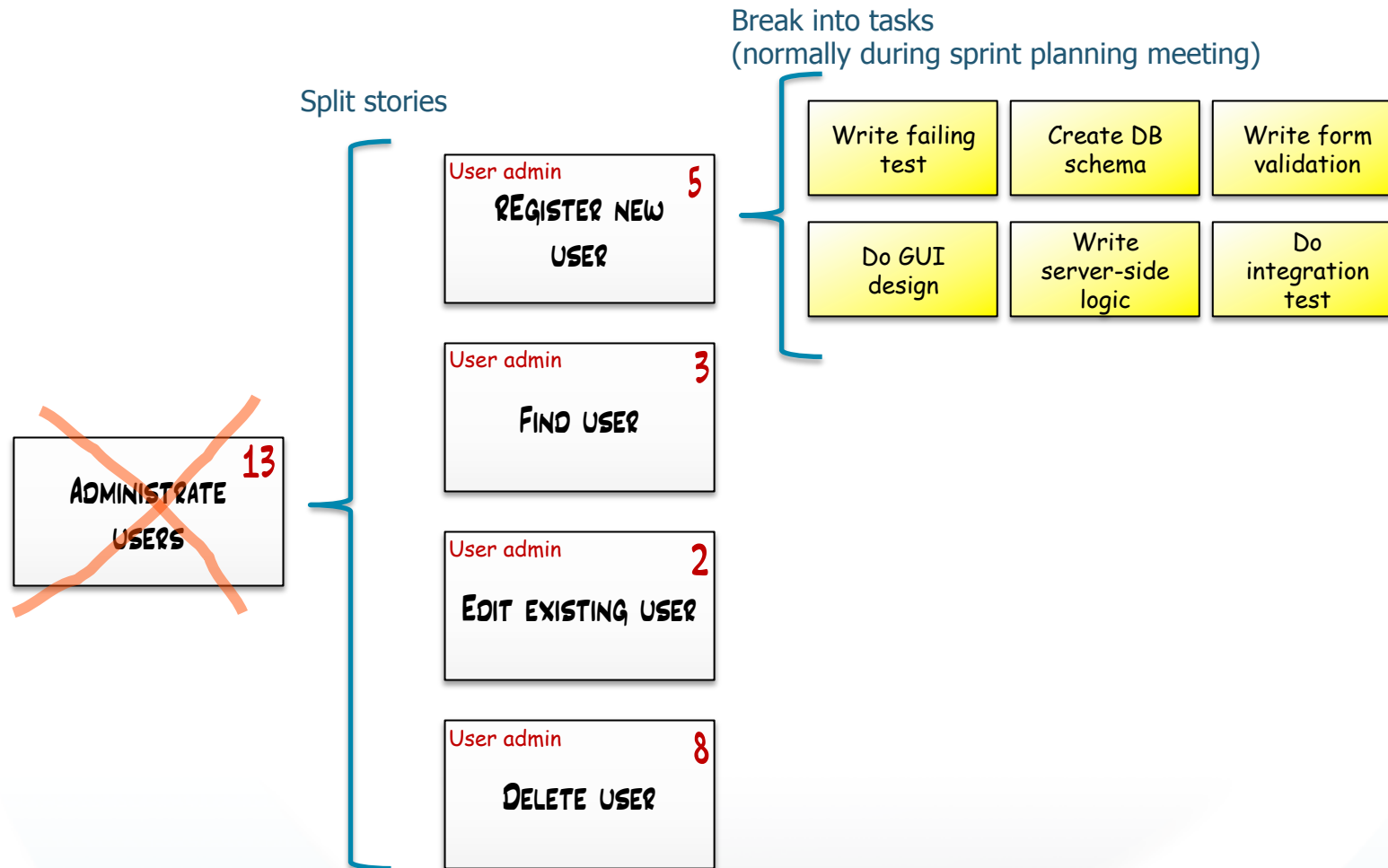
Yes!

Recommendation #8:

Don't estimate tasks

28

Stories vs tasks



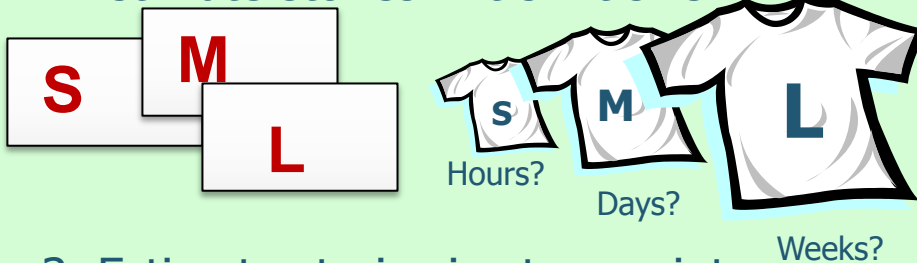
Estimation alternatives

Stories

1. Don't estimate stories. Just count them.



2. Estimate stories in t-shirt size



3. Estimate stories in story points



4. Estimate stories in ideal man-days



Henrik Kniberg

Tasks

1. Skip tasks

Recommendation:

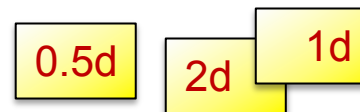
Don't estimate tasks



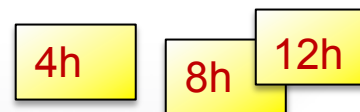
2. Don't estimate tasks. Just count them.



3. Estimate tasks in days



4. Estimate tasks in hours



Recommendation #9:

Skip focus factor

31

Skip Focus Factor

Let's say last sprint completed 18 story points using a 3-person team consisting of Tom, Lisa, and Sam working 3 weeks for a total of 45 man-days. And now we are trying to figure out our estimated velocity for the upcoming sprint. To complicate things, a new guy Dave is joining the team for that sprint. Taking vacations and stuff into account we have 50 man-days next sprint.

LAST SPRINT'S FOCUS FACTOR:

$$40\% = \frac{18 \text{ STORY POINTS}}{45 \text{ MAN-DAYS}}$$

THIS SPRINT'S ESTIMATED VELOCITY:

$$50 \text{ MAN-DAYS} \times 40\% = 20 \text{ STORY POINTS}$$



So our estimated velocity for the upcoming sprint is 20 story points. That means the team should add stories to the sprint until it adds up to approximately 20.

Focus Factor MAY be useful when starting a new project with a new team...

Facts:

- Team size = 6 people
- Sprint length = 2 weeks

Estimate:

- 2 sp = 6 imd
- Focus factor: 50%

Calculation:

- Calendar days in sprint: 10 days
- Man-days in sprint: 60 md
- Ideal man-days in sprint: $60 \text{ md} \times 50\% = 30 \text{ imd}$
- Velocity: $30 \text{ imd} / 3 = 10 \text{ sp / sprint}$

As A X
I WANT Y
S THAT Z 8sp

As A X
I WANT Y
S THAT Z 2sp

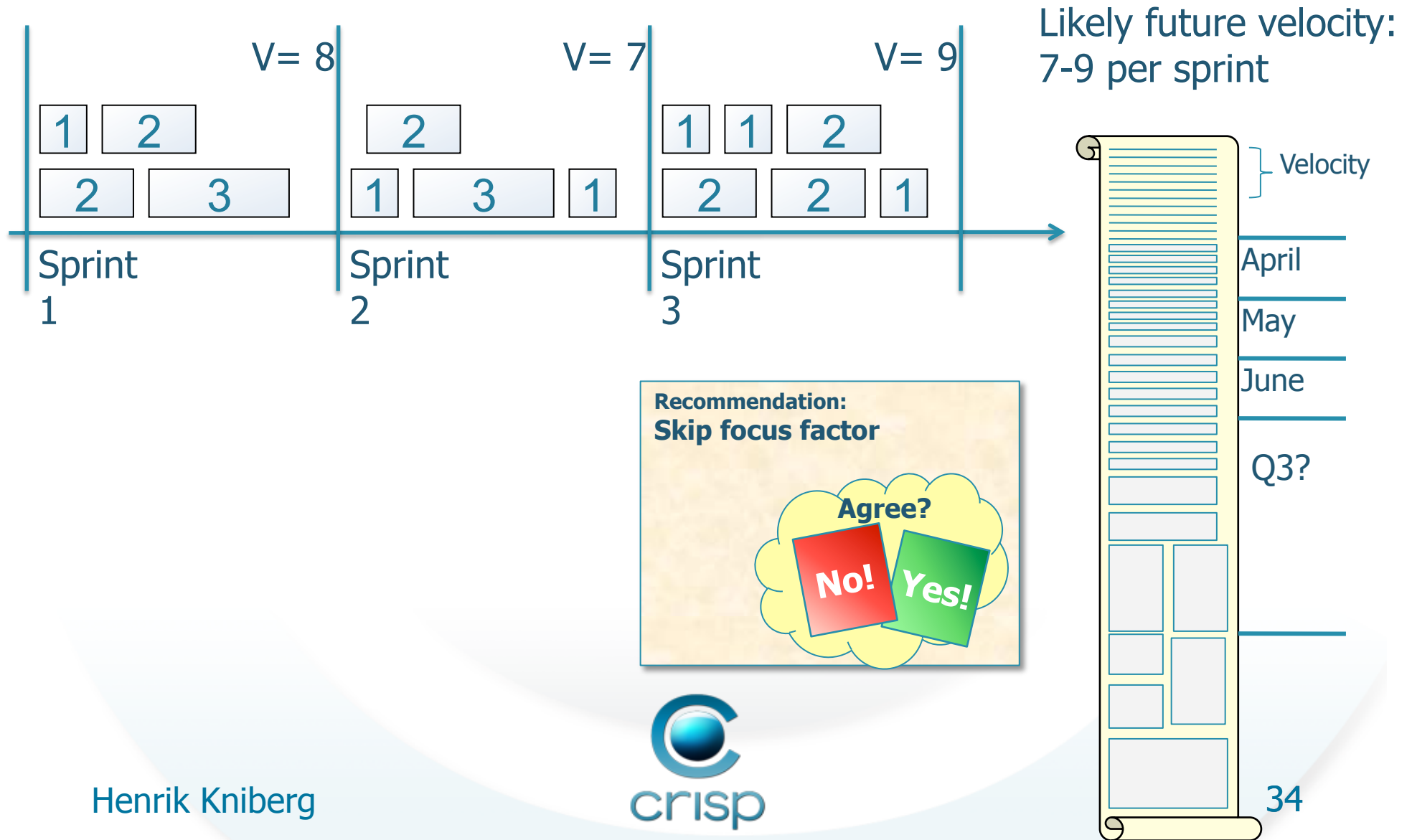
As A X
I WANT Y
S THAT Z 3sp

As A X
I WANT Y
S THAT Z 5sp

Hmmm.... we think THIS story is about 6 ideal man-days



But “simple” velocity calculations are simpler and usually no less accurate



Recommendation #10:

Don't fill the sprint

35

The sprint commitment

Common misconceptions

- ~~"We promise to achieve this goal"~~
- ~~"We promise to deliver all stories included in the sprint backlog"~~

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 sprint goal, and will let you know immediately if we no longer believe we can reach it.
- ... we believe that we can complete all stories included in the sprint backlog.
- ... we will demonstrate releasable code at the end of the sprint
- ... if we fall behind schedule we will talk to you and, if necessary, remove the lowest priority stories first.
- ... if we get ahead of schedule, we will add stories to the sprint from the product backlog, in priority order.
- ... we will display our progress and status on a daily basis.
- ... every story that we do deliver is *Done*.

11-02-16

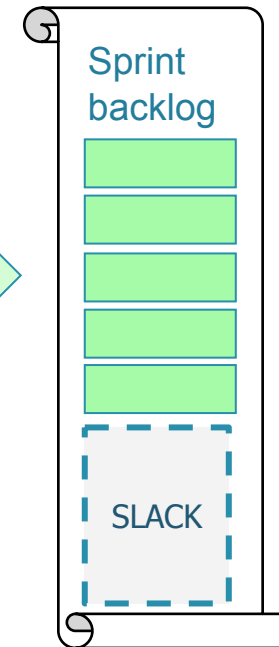
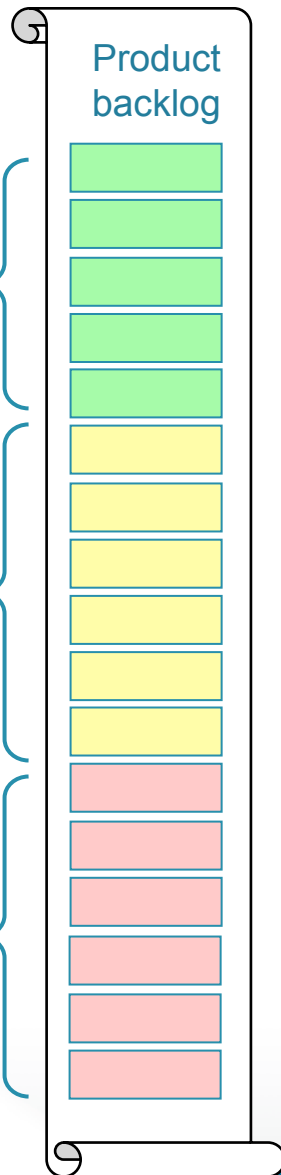
Henrik Kniberg



- **Team:** We're pretty sure we can finish all of these.
- **PO:** These are definitely the highest priority items, their priorities won't change during the sprint.

- **Team:** We might finish some of these, but probably not all
- **PO:** Some of these might be reprioritized during the sprint

- **Team:** We're pretty sure we won't finish any of these



Recommendation:
Don't fill the sprint

Agree?

No!

Yes!

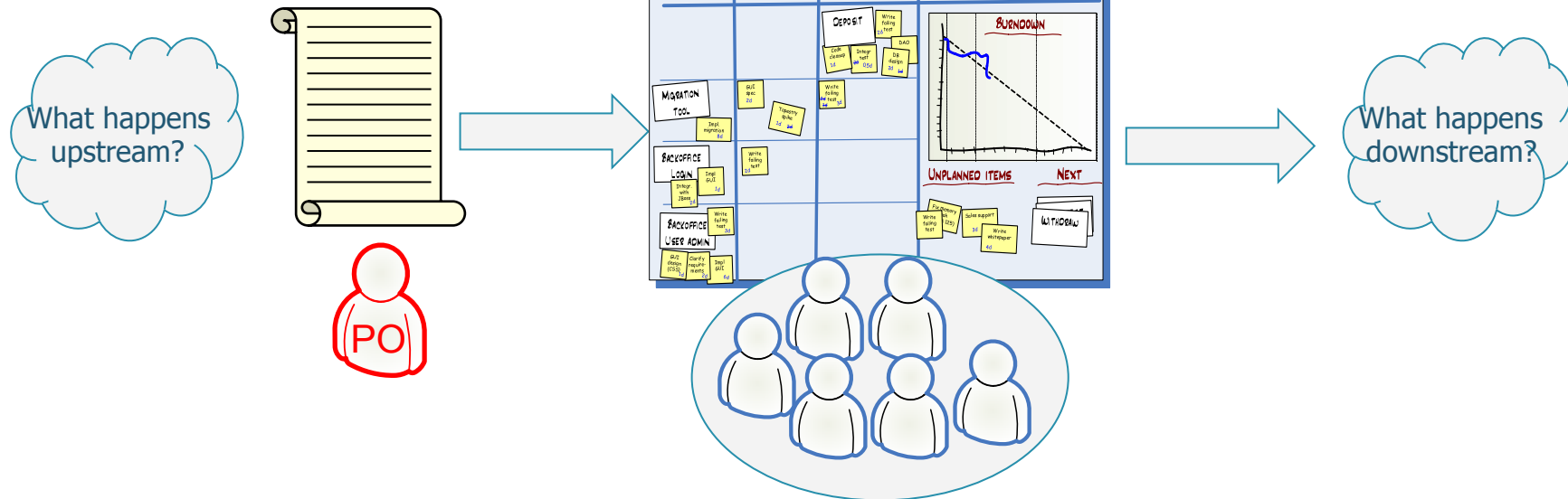
Recommendation #11:

**Visualize more than
just the sprint**

38

Product backlog

Sprint



Analysis

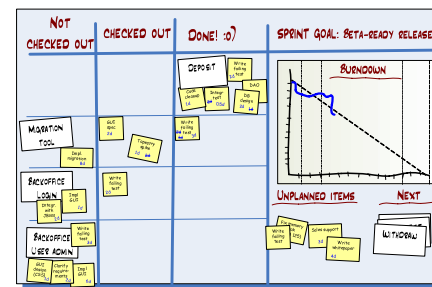
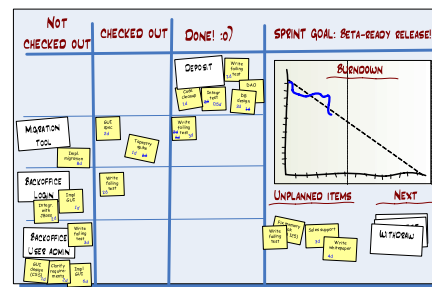
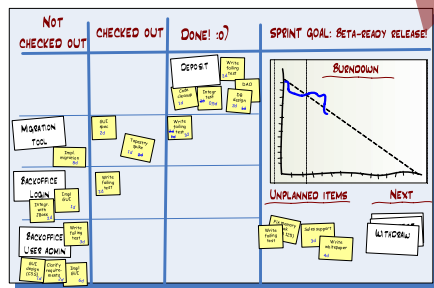
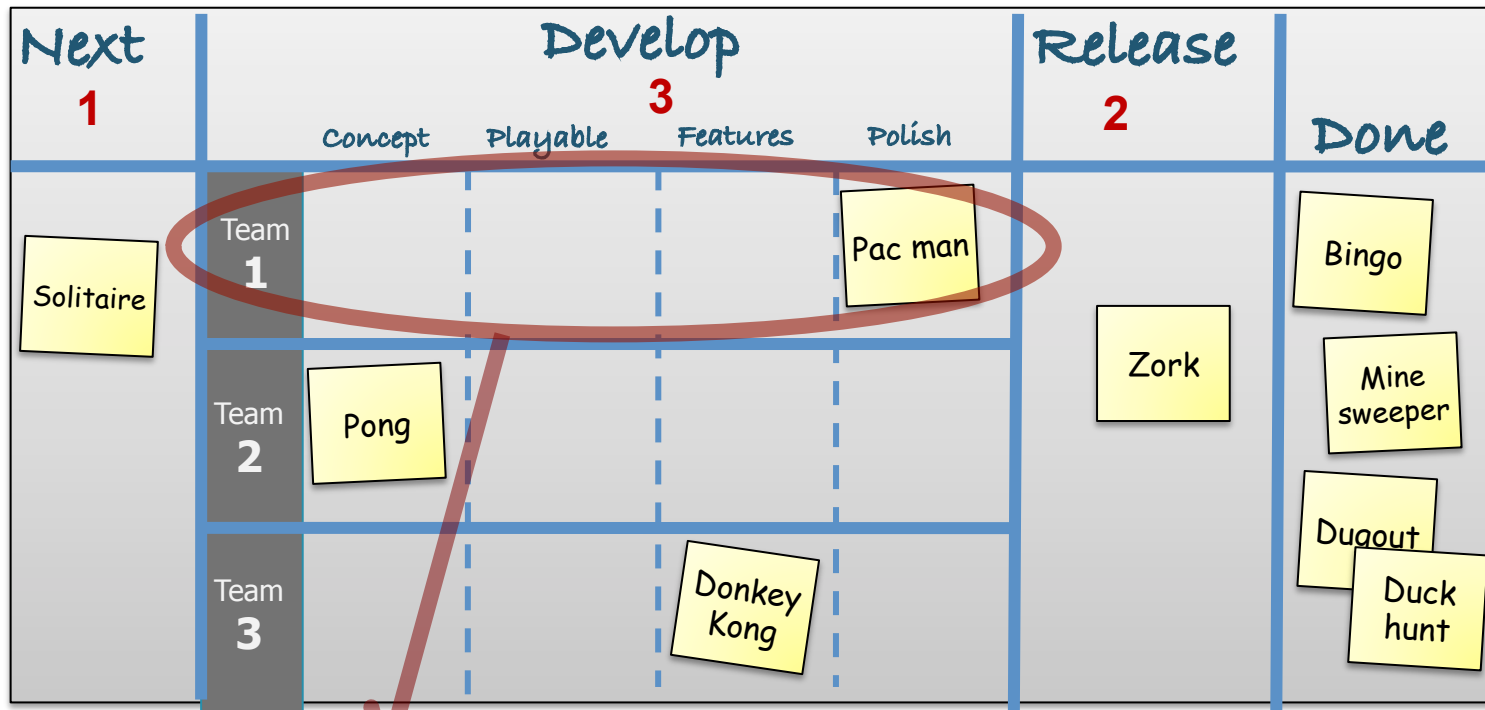
Product
backlog

Development

User acceptance test
& production

Support





Recommendation:
Visualize more than just the sprint

Agree?

No!

Yes!

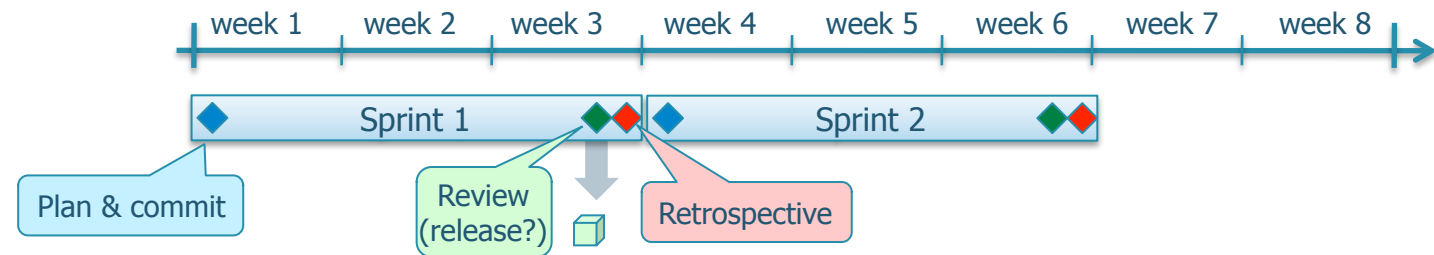
Recommendation #12:

**Use Kanban when
sprints don't work**

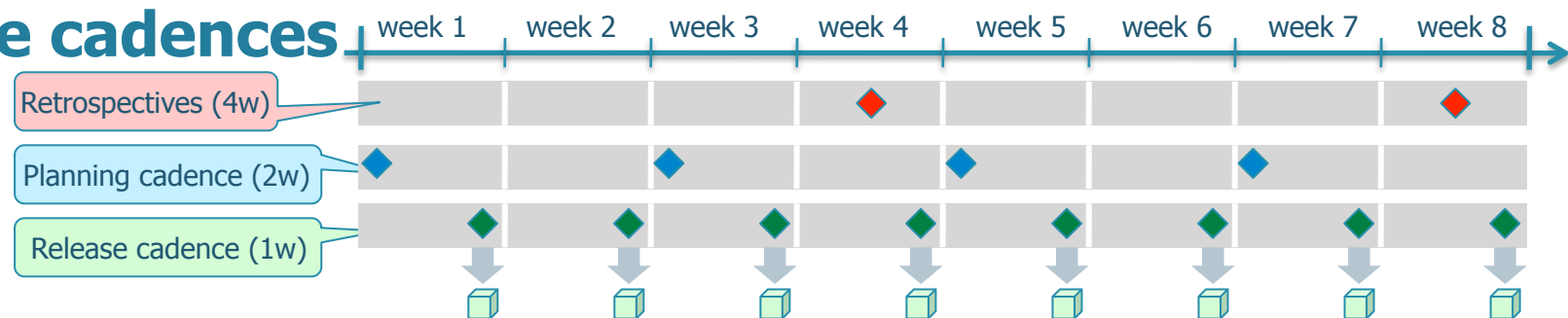
42

Sprints aren't the only way to be agile

Sprints



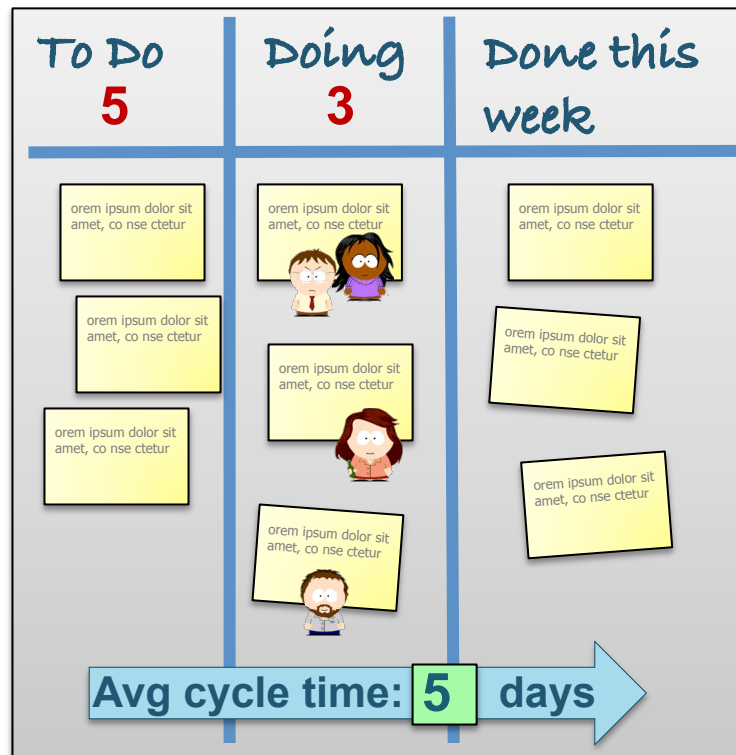
Separate cadences



Event-driven



A simple kanban system



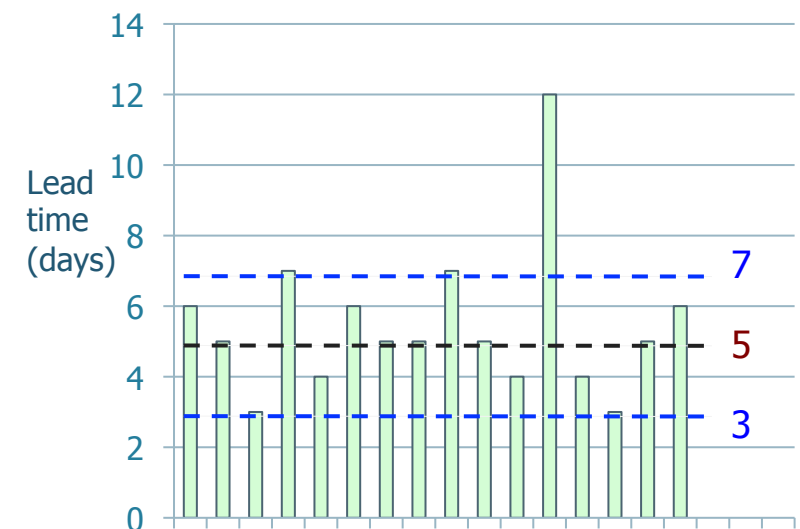
w1	w2	w3	w4	w5	w6		
8	10	7	7	9			



Henrik Kniberg

2011-01-05 2011-01-09

Migrate invoicing reports to the new format



Recommendation:
Use Kanban when sprints don't work



Recommendation #13:

**Use the “king & servant”
pattern to enable
continuous delivery**

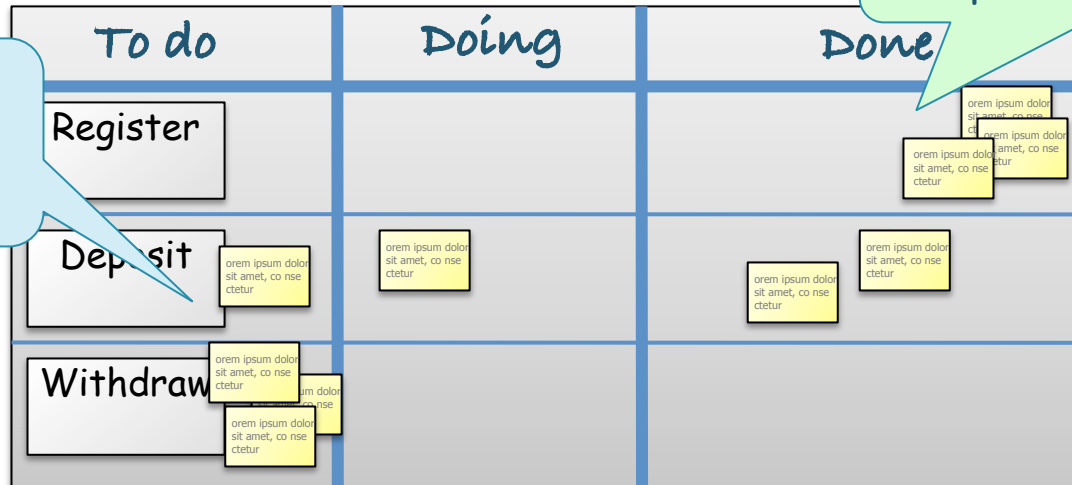
45

Parallellitis (n)

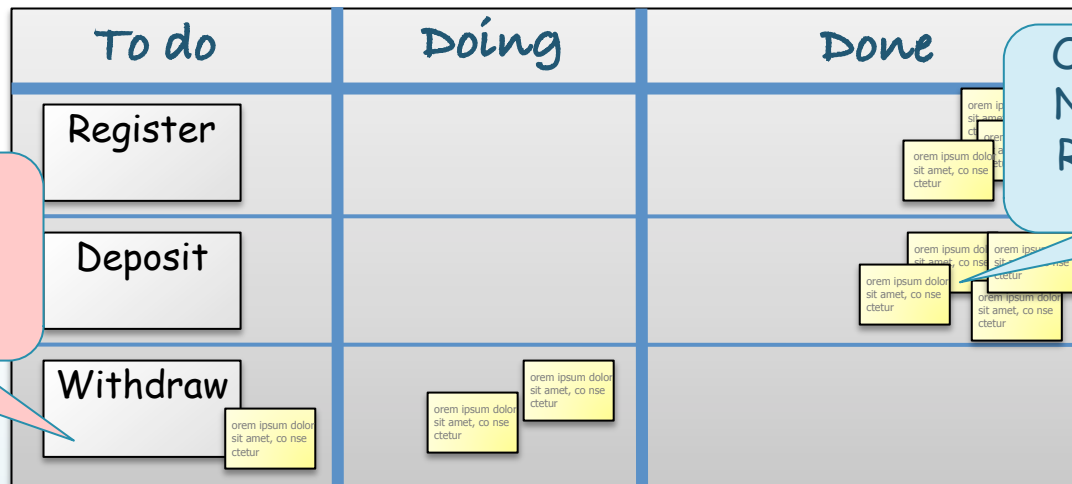
The mistaken belief that you get more done by doing everything at once.

Register is done! Let's publish to trunk!

No wait, there is partially complete Deposit code! Lemee finish!



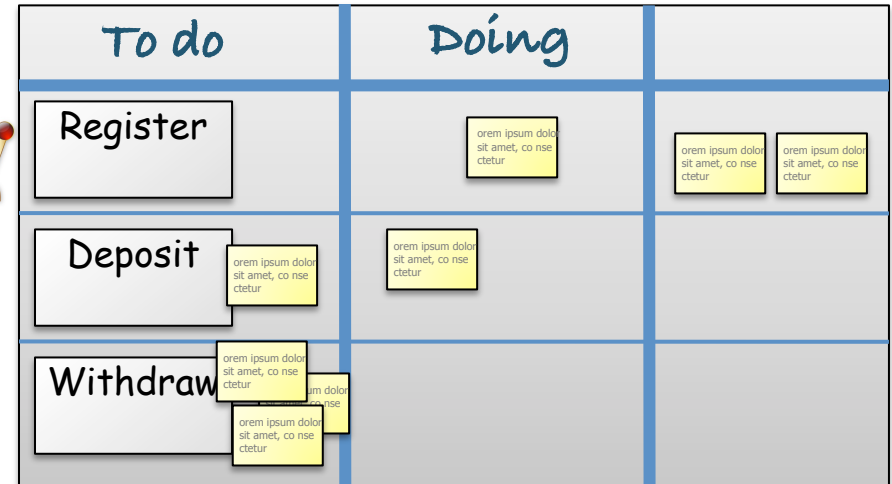
No, wait! There is partially complete Withdraw code. Lemee finish!



OK Deposit is done! Now we can publish Register & Deposit to trunk!

The king & servant pattern

- Anyone working on the top priority story is King.
- Everyone else on the team is a Servant.
- You want to be King. Try to find ways to help out with the top priority story.
- Whenever a King needs help, Servants immediately offer their services.
- A Servant may not disrupt a King.
- A Servant may never check in unreleasable code on the team branch. A King may check in whatever he pleases (as long as he doesn't violate a branch policy of course).
- As soon as the top priority story is Done, anyone working on the next story is now King.



Recommendation:
Use the "King & servant"
pattern to enable
continuous delivery

Agree?

No!

Yes!

"Version Control for Multiple Agile Teams"

<http://www.infoq.com/articles/agile-version-control>

InfoQ
490,924 Sep unique visitors

Architectures you've
always wondered about
NOV 1-5 2010 SAN FRANCISCO

Register
Login
About us
Personal feed
QCon

Your Communities

- ☐ Java
- ☐ .NET
- ☒ Ruby
- ☐ SOA
- ☐ Agile
- ☐ Architecture
- ☐ Operations

Tracking change and innovation in the enterprise software development community

Article

Version Control for Multiple Agile Teams

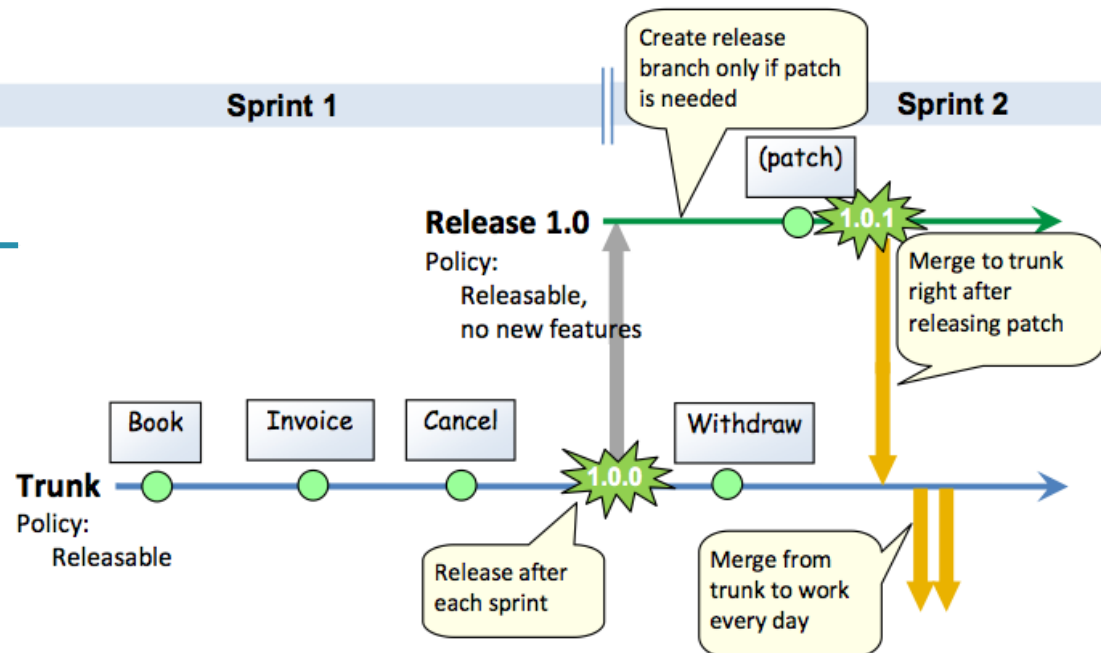
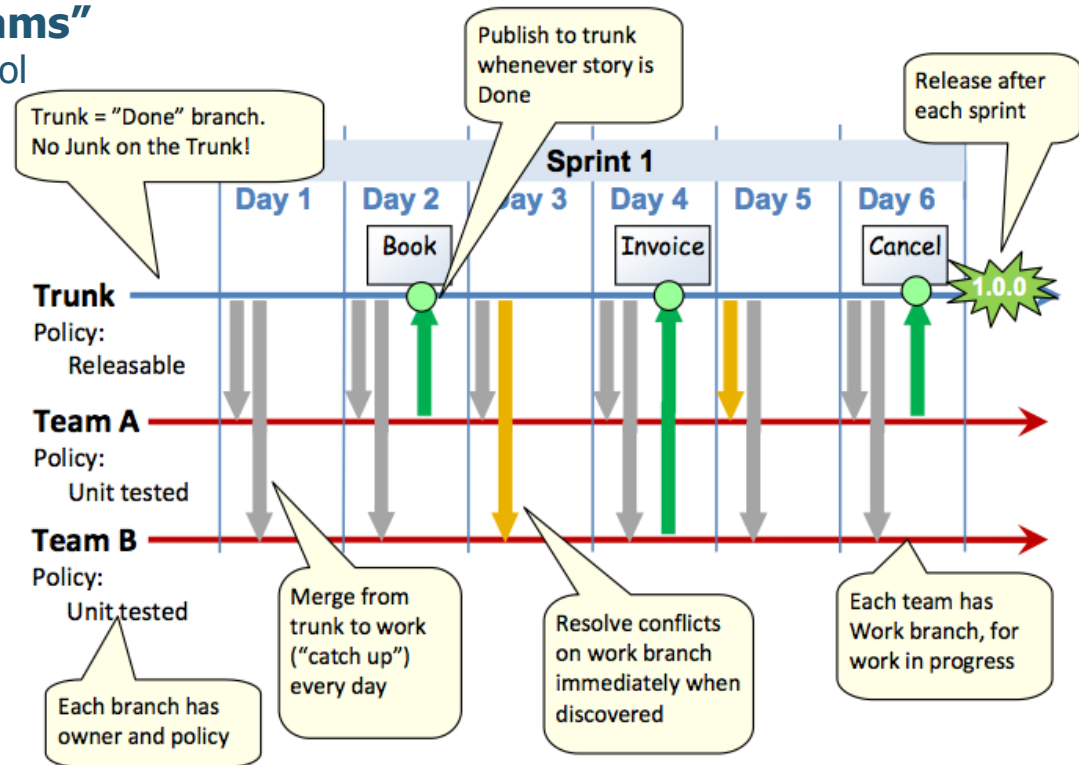
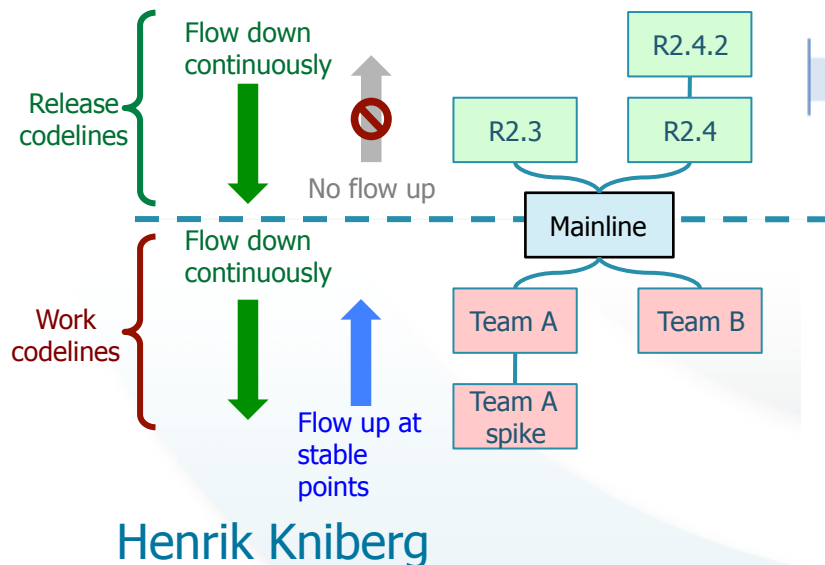
Posted by **Henrik Kniberg** on Mar 31, 2008

Community Agile Topics Configuration Management, Artifacts & Tools Tags Releases, Best Practices, Automation

Share

If we have several agile development teams working on the same codebase, how do we minimize the risk of stumbling over each other? How do we ensure that there always is a clean, releasable version at the end of each iteration? This paper describes an example of how to handle version control in an agile environment with multiple teams – it is the scheme that we migrated to at the company described in "[Scrum and XP from the Trenches](#)".

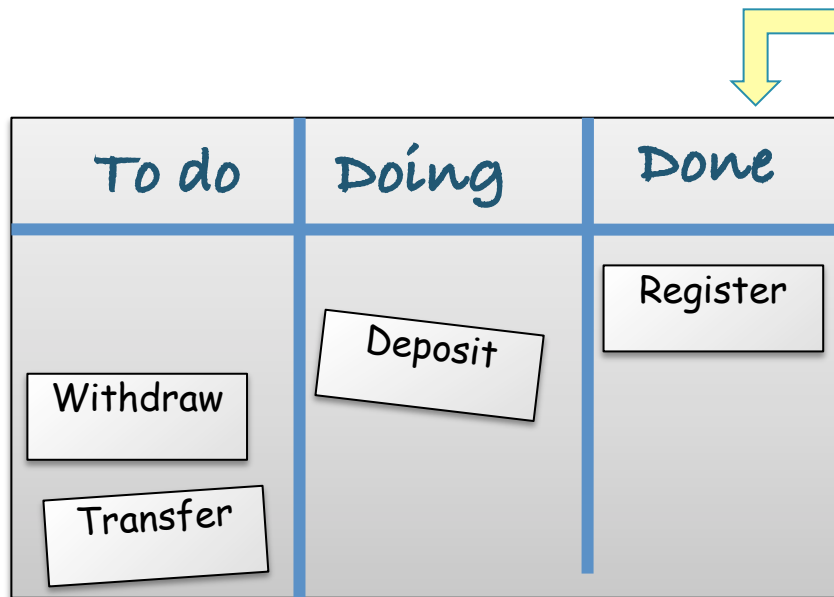
This paper is not primarily targeted for version control experts, in fact such experts probably won't find anything new here. This paper is aimed at the rest of us, those of us that just want to learn simple and useful ways to collaborate. It may be of interest to anyone directly involved in agile software development, regardless of role – branching and merging is everybody's business, not just the configuration manager.



Recommendation #14:

**Use Definition of
Done to manage
technical debt**

49



Definition of Done

Releasable

- Merged to trunk
- Acceptance tested
- No increased technical debt

= I haven't messed up the codebase

Recommendation:

Use Definition of Done to manage technical debt



Henrik Kniberg

Simple design

1. Passes all tests
2. No duplication
3. Readable
4. Minimal



Kent Beck



Recommendation #15:

Test automation backlog

51

Step 1: Decide what needs to be tested

- Change skin
- Security alert
- Transaction history
- Block account
- Add new user
- Sort query results
- Deposit cash
- Validate transfer

Step 2: Classify each test

Pay every
time

Pay once

Test case	Risk	Manual Test Cost	Automation Cost
Change skin	low	0.5 hrs	20 sp
Security alert	high	1 hrs	13 sp
Transaction history	med	3 hrs	1 sp
Block account	high	5 hrs	0.5 sp
Add new user	low	0.5 hrs	3 sp
Sort query results	med	2 hrs	8 sp
Deposit cash	high	1.5 hrs	1 sp
Validate transfer	high	3 hrs	5 sp

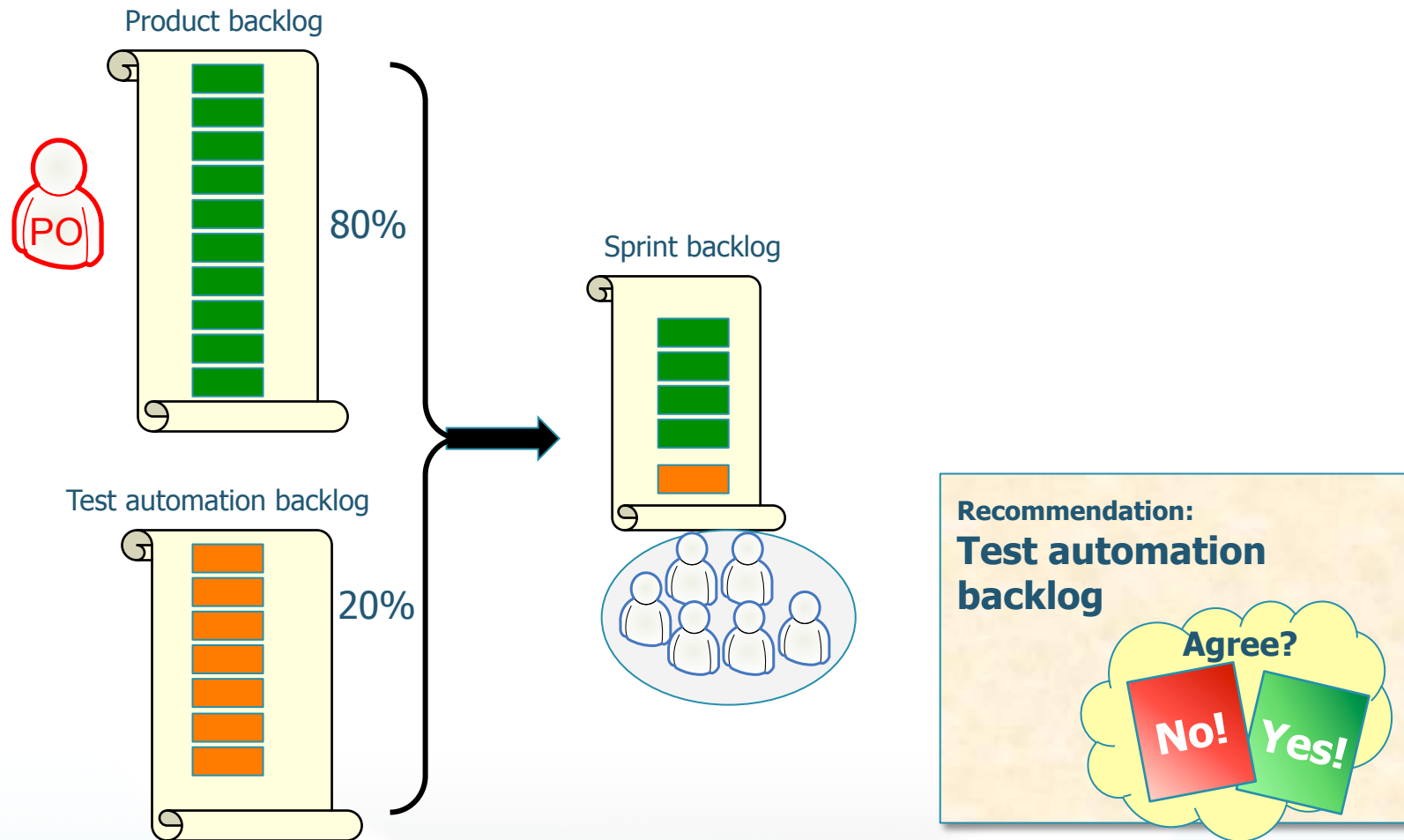
Step 3: Sort the list

Automate first!

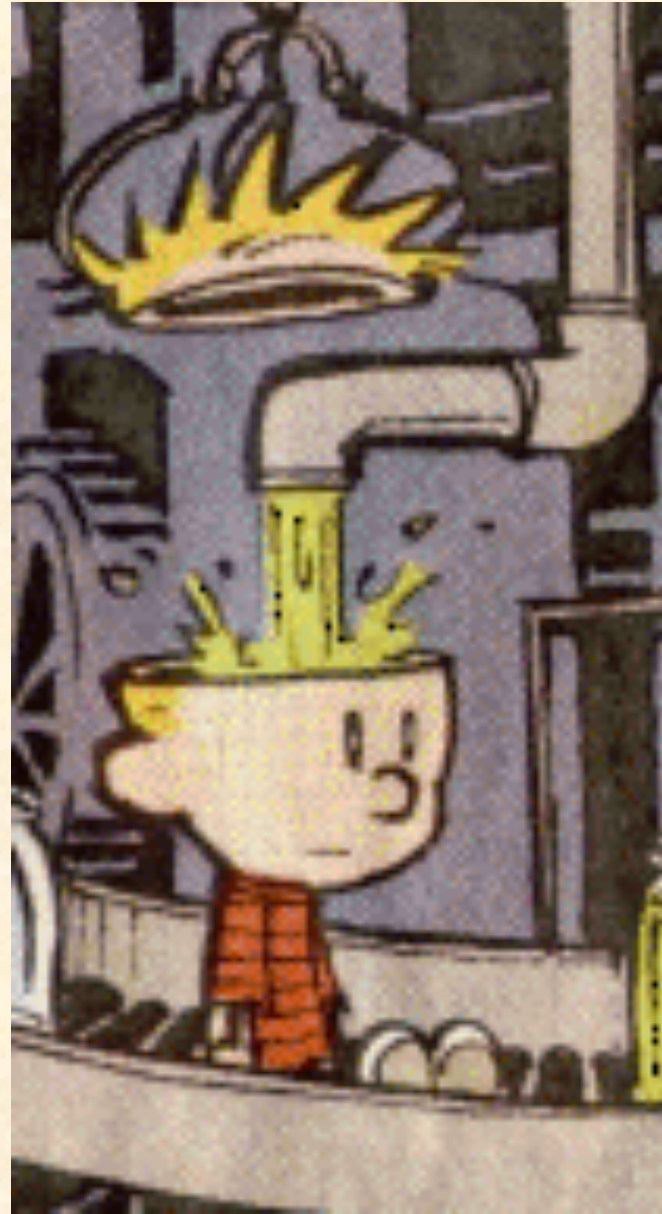
Test case	Risk	Manual Test Cost	Automation Cost
Block account	high	5 hrs	0.5 sp
Validate transfer	high	3 hrs	5 sp
Transaction history	med	3 hrs	1 sp
Sort query results	med	2 hrs	8 sp
Deposit cash	high	1.5 hrs	1 sp
Security alert	high	1 hr	13 sp
Add new user	low	0.5 hrs	3 sp
Change skin	low	0.5 hrs	20 sp

Automate last (or never)!

Step 4: Allocate capacity



Wrapup



Perfection is a direction, not a place



- Find a balance: appreciate your current solution, while looking for a better one.
- Just because today's solution seems better, doesn't mean yesterday's solution was bad.
- Copy/pasting solutions from other contexts is OK, as long as you also adapt.
- Keep experimenting & make sure the learning never stops.

