Agile Everywhere!

Keynote, Agile Tour Montreal
Nov 16, 2016

Henrik Kniberg
henrik.kniberg@crisp.se
@HenrikKniberg

Consultant

crisp
www.crisp.se

Dad

Organizational coach & Change Instigator

Spotify

LEGO

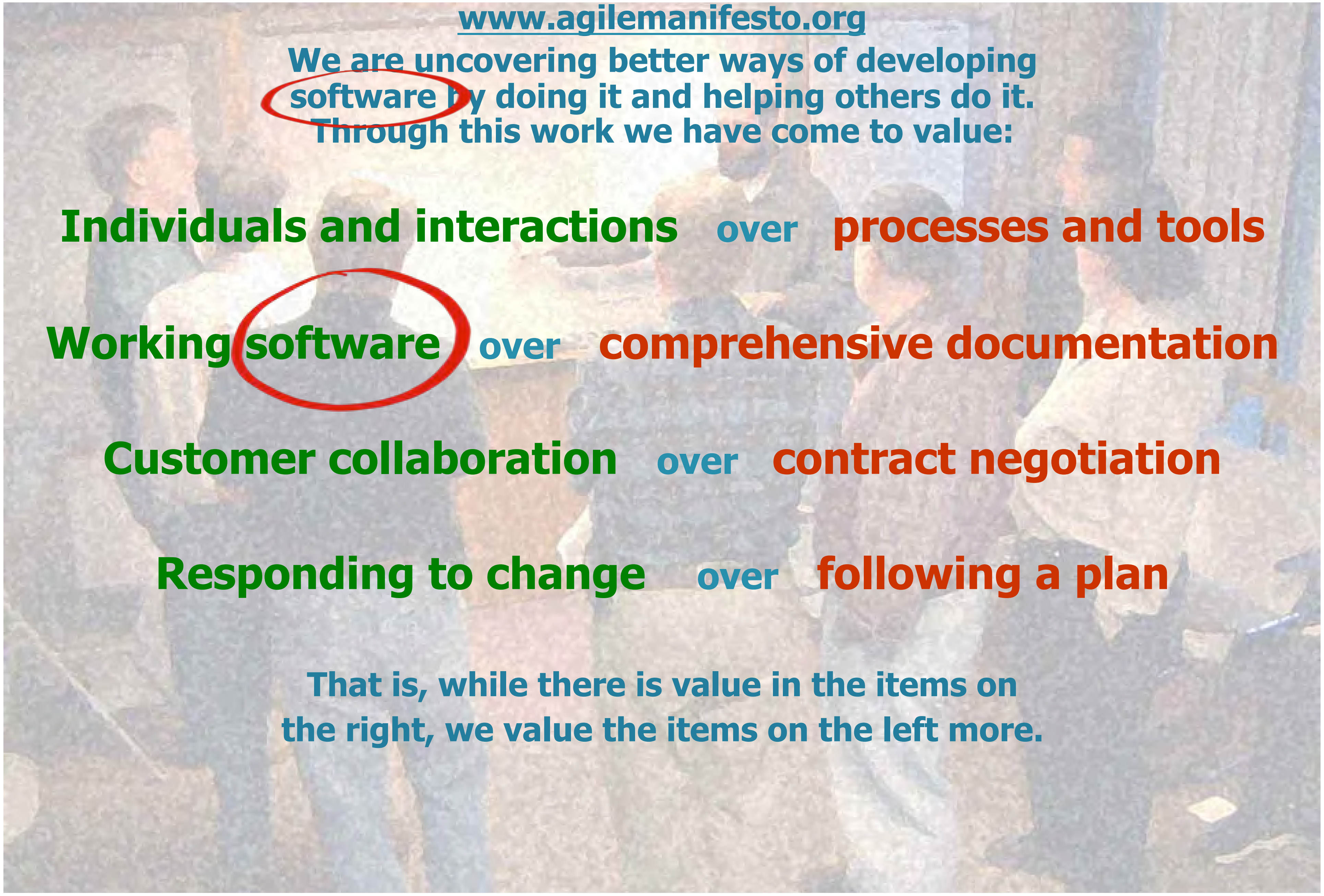
Author

Scrum and XP from the Trenches
Henrik Kniberg
InfoQ

Kanban and Scrum making the most of both
Henrik Kniberg & Mattias Skarin
InfoQ

LEAN FROM THE TRENCHES
Henrik Kniberg

Is Agile just a Software thing?



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.

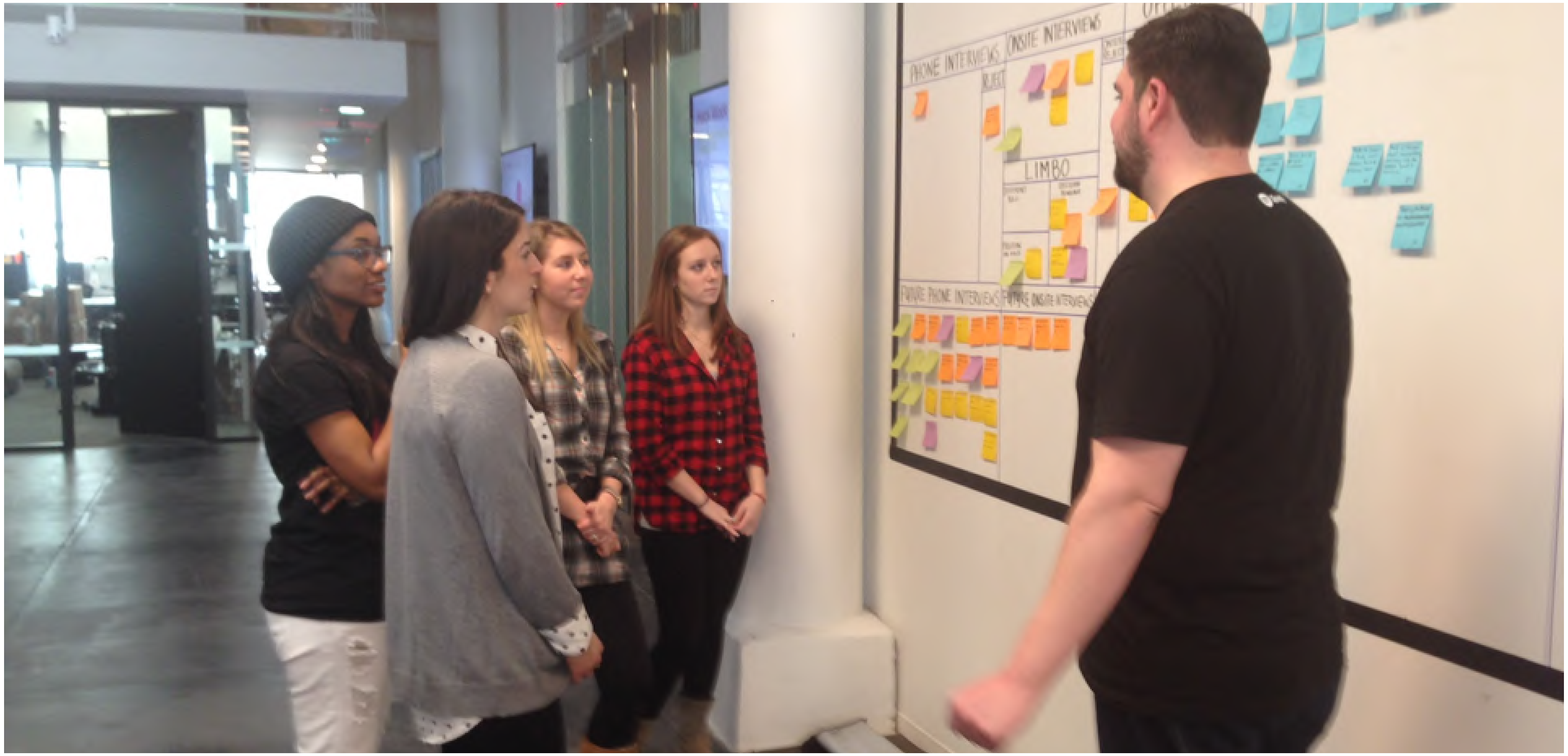
Henrik Kniberg

BBQ Board
To do
Going on
Done!
Drink table
Dip snacks
Light the BBQ
Orange Juice
Veggie sticks
Sallad
Food serving table
BBQ
Ice cream
Grapes + cheese crackers

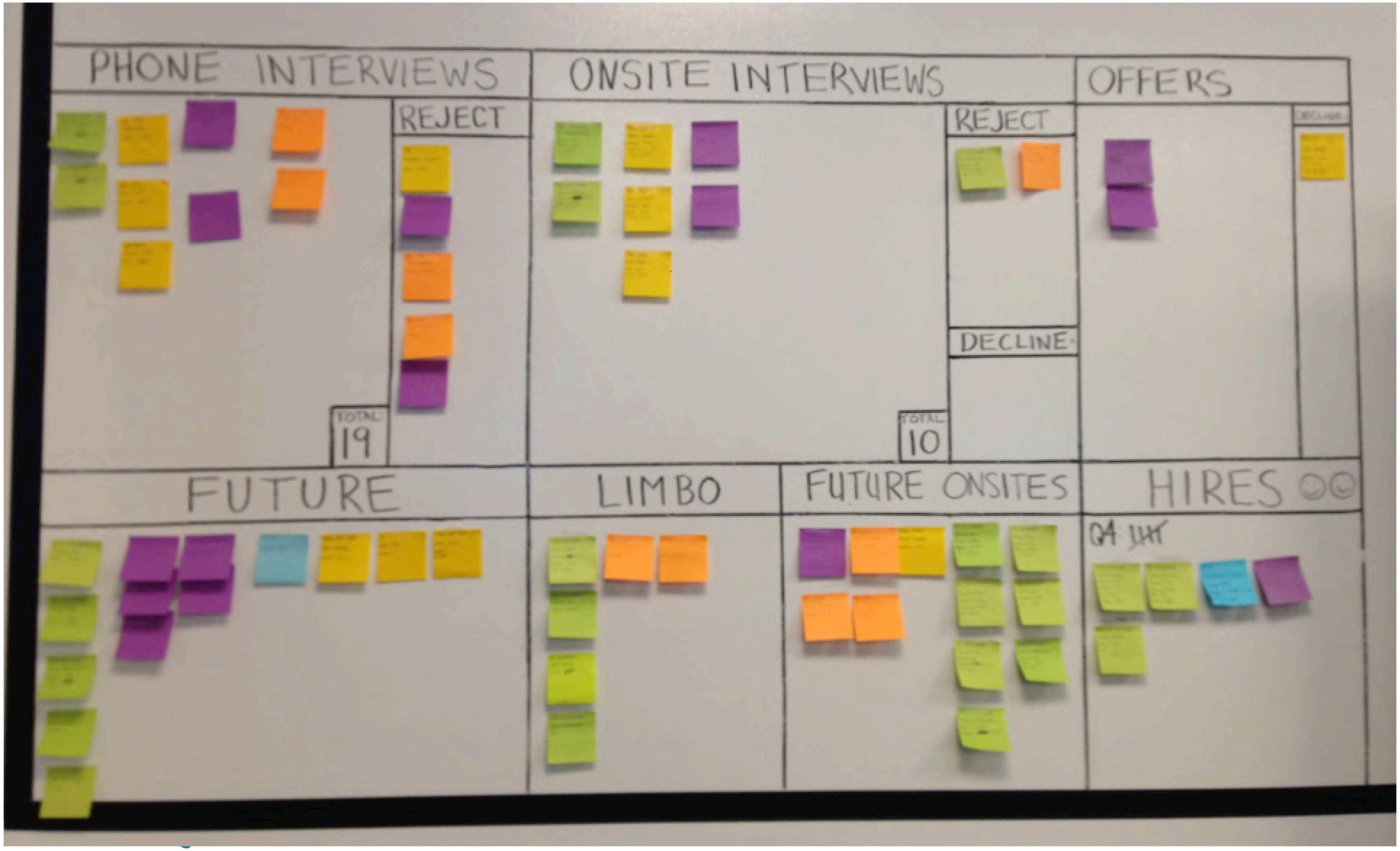
SKATT

16 ♀
X

Recruitment team



Recruitment team



JAS 39E Saab Gripen

World's most cost-effective military aircraft
($4700 Cost per Flight Hour)

Agile practices implemented at every level and in every discipline: software, hardware and fuselage design.

Compared to F35 joint strike fighter, Gripen 39E has:

- 50x lower development cost!
- 10x lower unit cost!

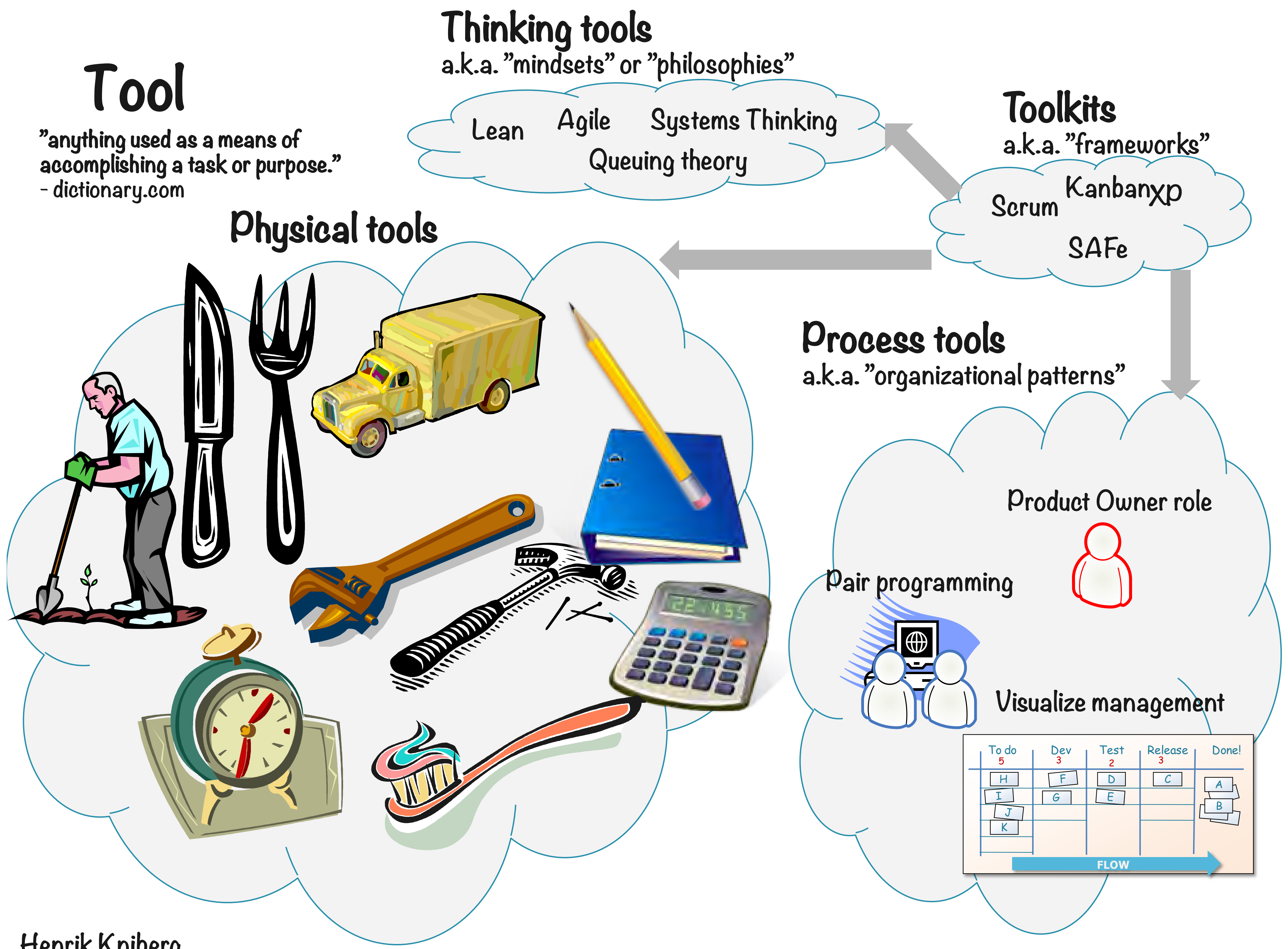
Pilots on the same site as development teams. Direct feedback provided every sprint.

1500 people, all co-located in Linköping, Sweden.

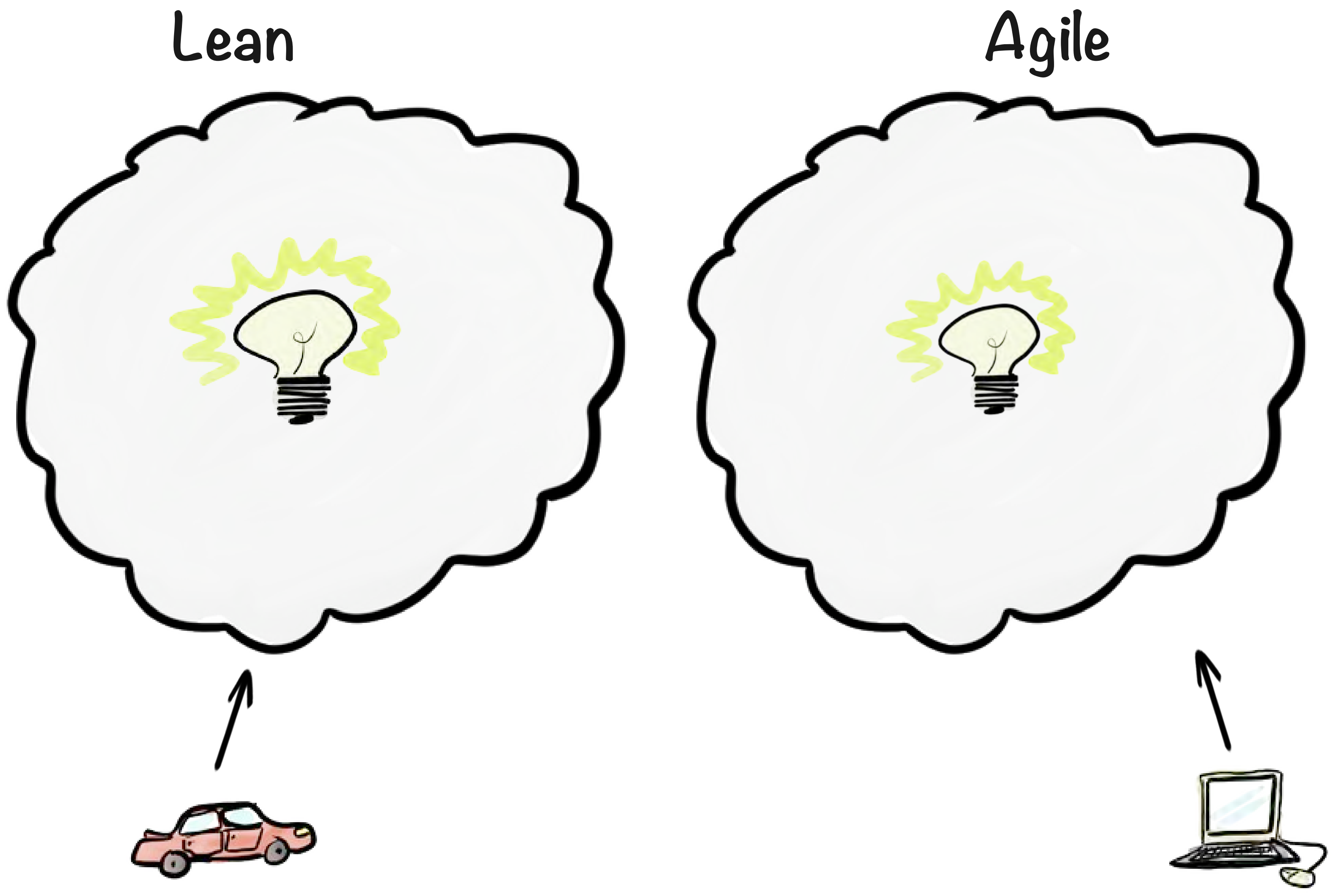
Sources:

- http://www.stratpost.com/gripen-operational-cost-lowest-of-all-western-fighters-janes
- Personal visit to SAAB Linköping
- Research paper "Owning the Sky with Agile"

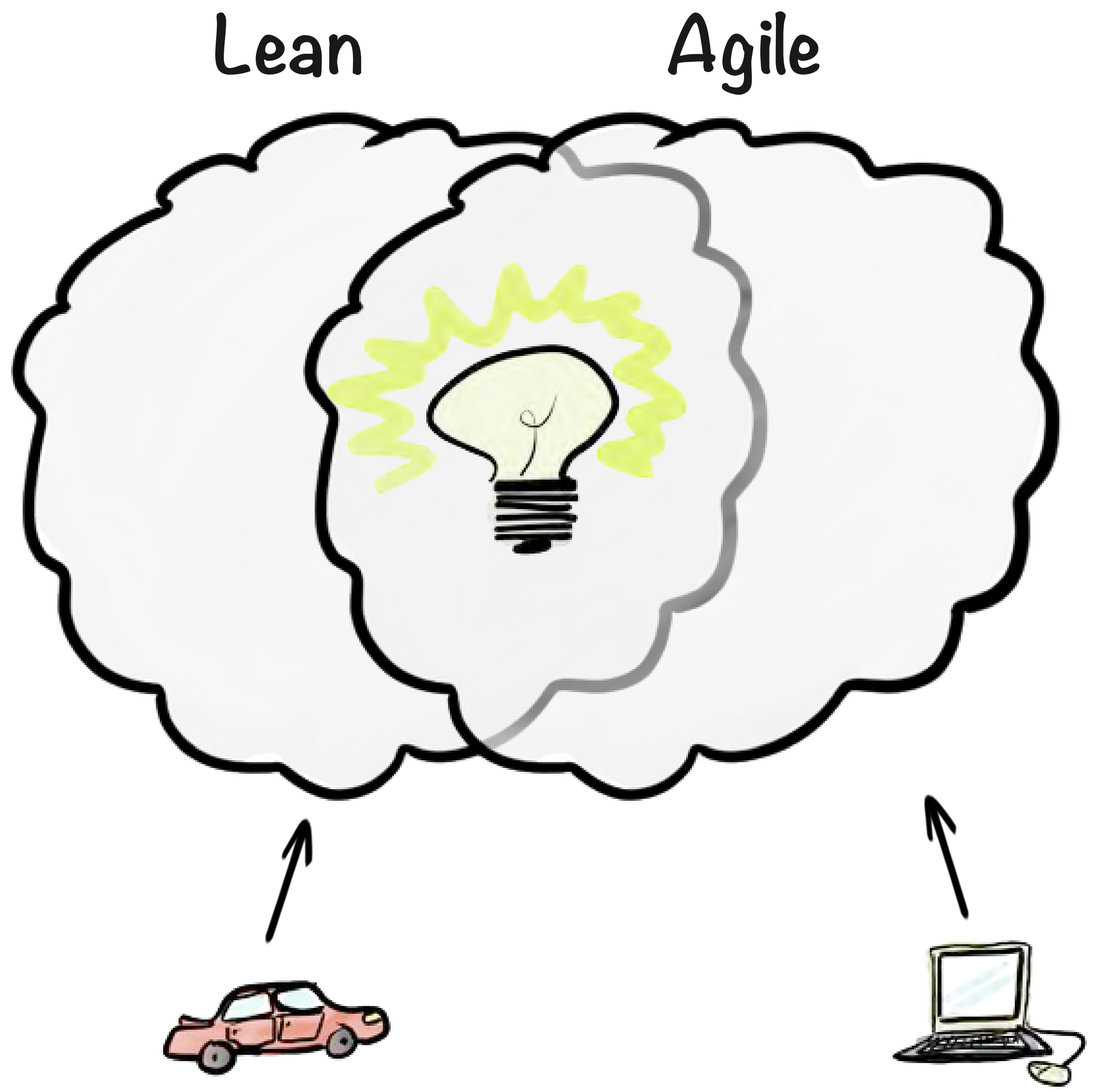
Henrik Kniberg



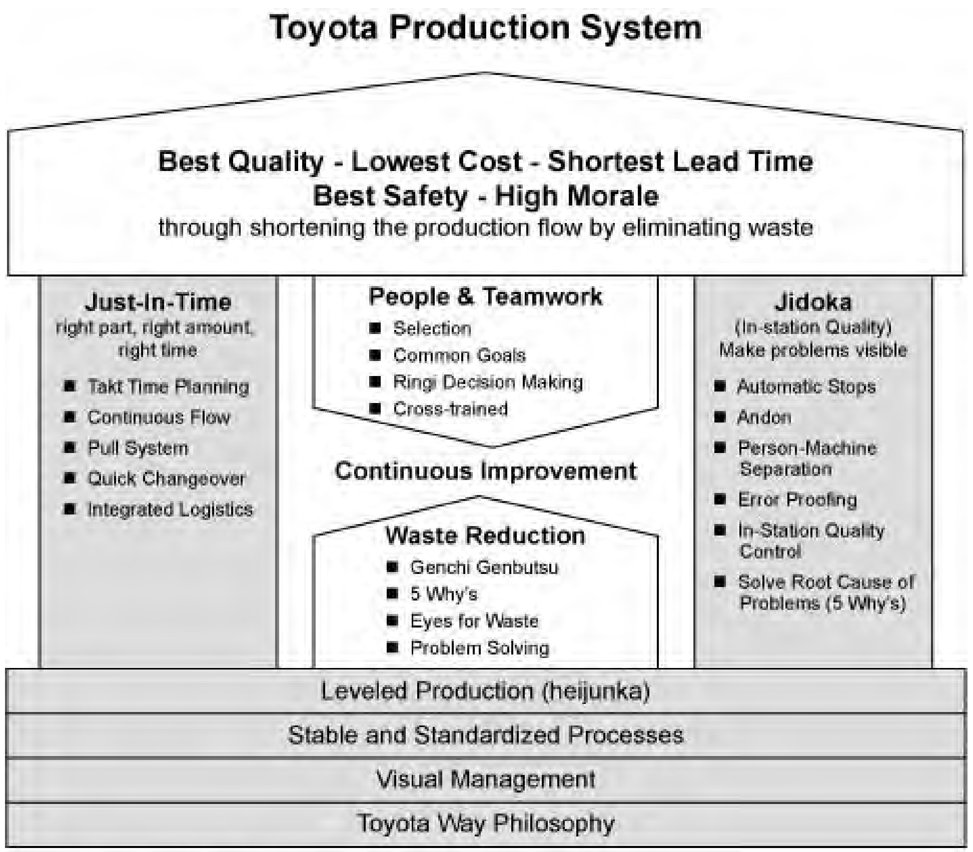
Henrik Kniberg



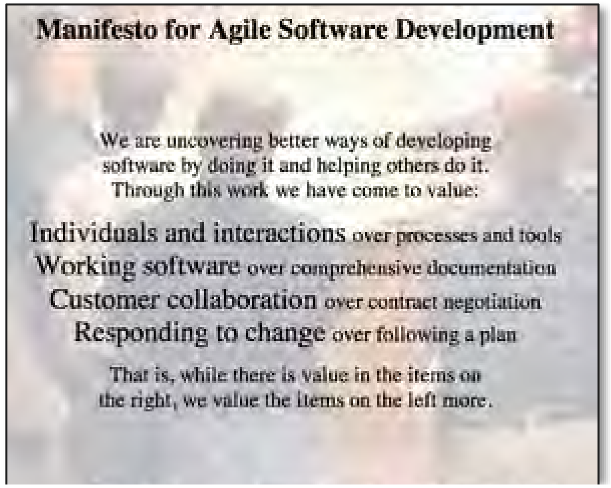
Henrik Kniberg



Henrik Kniberg



Source: J. Liker (2004). The Toyota Way. McGraw-Hill. pg. 33.

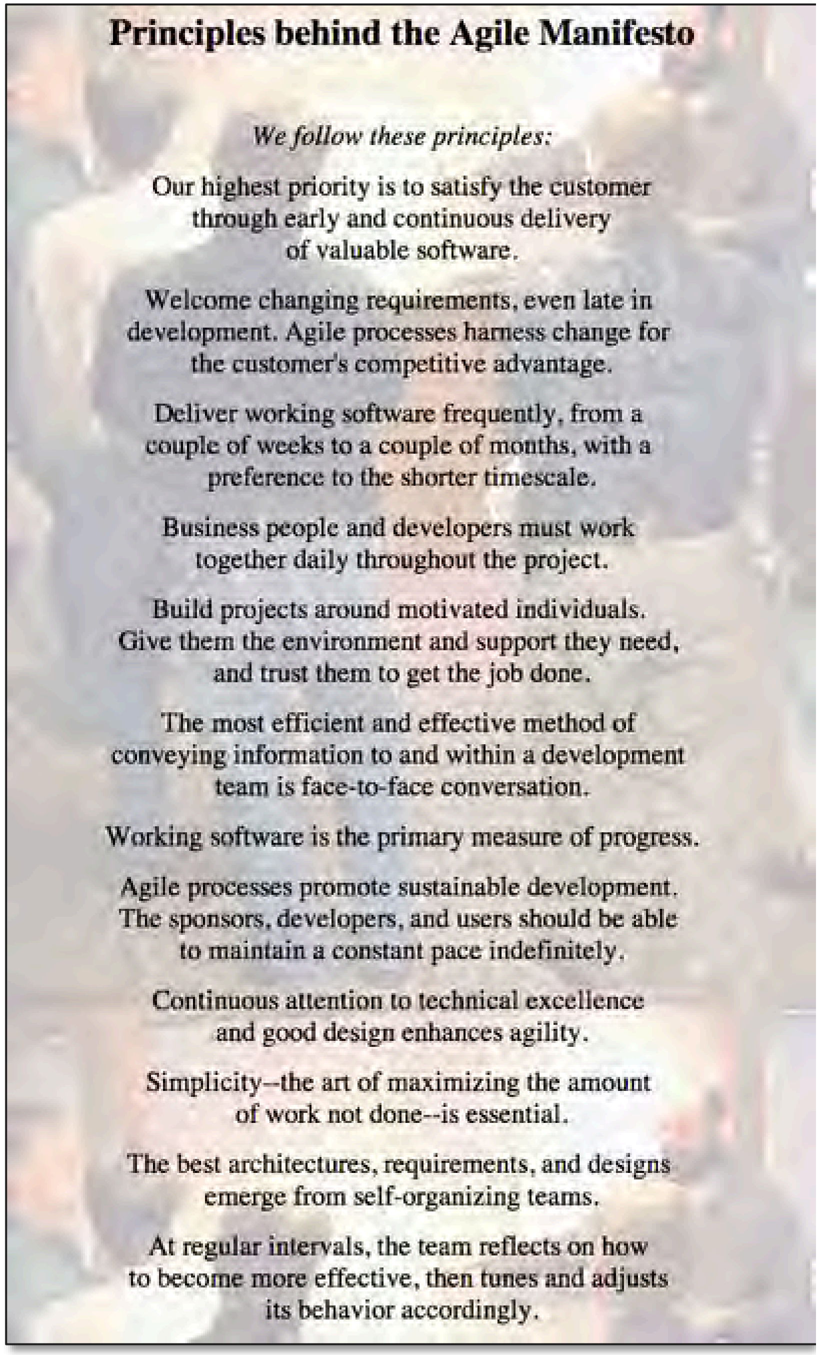


Manifesto for Agile Software Development

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.



Principles behind the Agile Manifesto

We follow these principles:

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

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

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

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

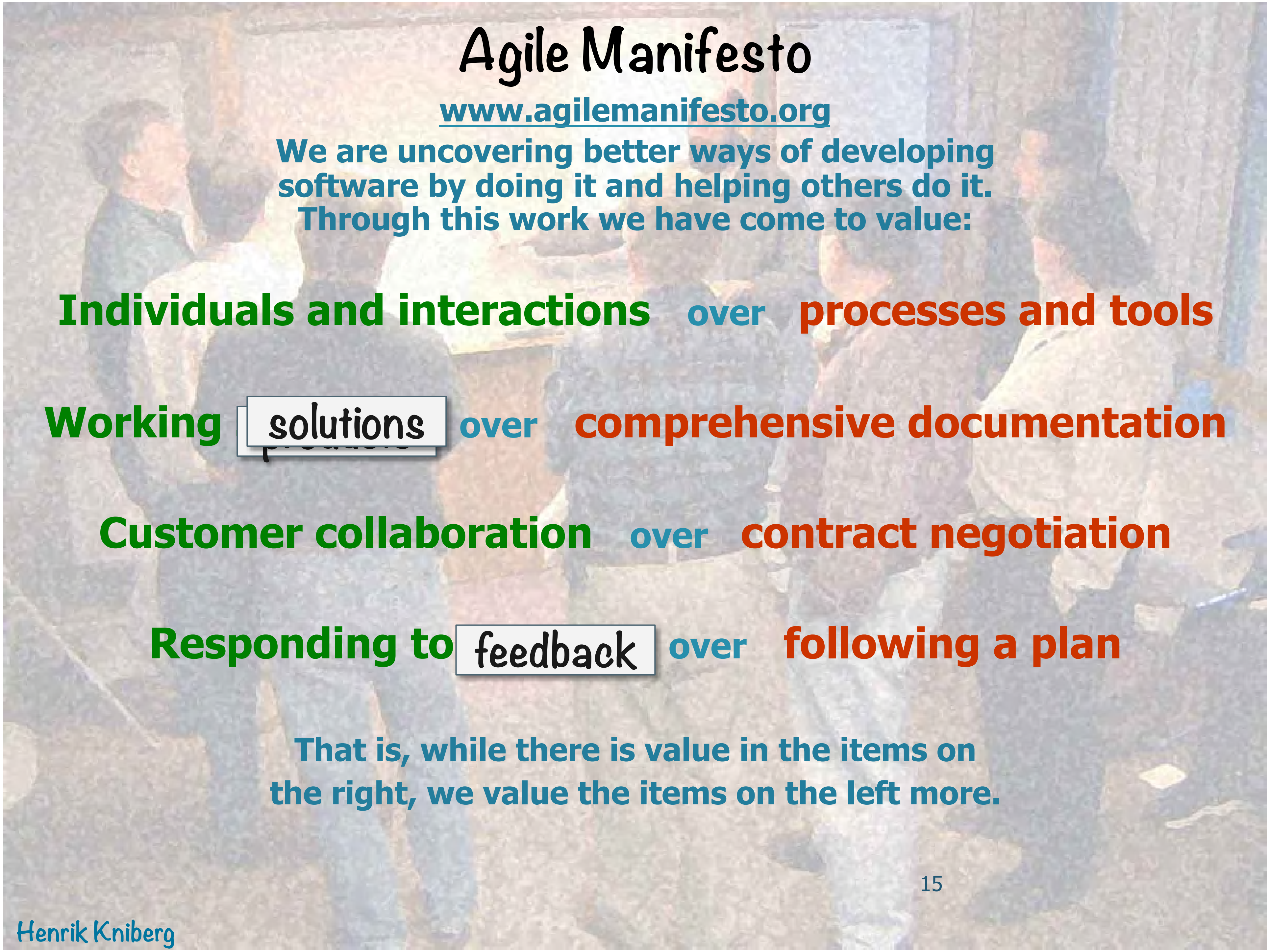
Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Henrik Kniberg



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 solutions over **comprehensive documentation**

Customer collaboration over **contract negotiation**

Responding to feedback over **following a plan**

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

Agile is not new

Buzzwords will come and go, but the underlying ideas and principles are timeless

COVER FEATURE

Iterative and Incremental Development: A Brief History

Although many view iterative and incremental development as a modern practice, its application dates as far back as the mid-1950s. Prominent software-engineering thought leaders from each succeeding decade supported IID practices, and many large projects used them successfully.

Craig Larman
Valtech

Victor R. Basili
University of Maryland

As agile methods become more popular, some view iterative, evolutionary, and incremental software development—a cornerstone of these methods—as the "modern" replacement of the waterfall model, but its practiced and published roots go back decades. Of course, many software-engineering students are aware of this, yet surprisingly, some commercial and government organizations still are not.

This description of projects and individual contributions provides compelling evidence of iterative and incremental development's (IID's) long existence. Many examples come from the 1970s and 1980s—the most active but least known part of IID's history. We are mindful that the idea of IID came independently from countless unnamed projects and the contributions of thousands and that this list is merely representative. We do not mean this article to diminish the unsung importance of other IID contributors.

We chose a chronology of IID projects and approaches rather than a deep comparative analysis. The methods varied in such aspects as iteration length and the use of time boxing. Some attempted significant up-front specification work followed by incremental time-boxed development, while others were more classically evolutionary and feedback driven. Despite their differences, however, all the approaches had a common theme—to avoid a single-pass sequential, document-driven, gated-step approach.

Finally, a note about our terminology: Although some prefer to reserve the phrase "iterative development" merely for rework, in modern agile methods the term implies not just revisiting work, but also evolutionary advancement—a usage that dates from at least 1968.

PRE-1970

IID grew from the 1930s work of Walter Shewhart,[1] a quality expert at Bell Labs who proposed a series of short "plan-do-study-act" (PDSA) cycles for quality improvement. Starting in the 1940s, quality guru W. Edwards Deming began vigorously promoting PDSA, which he later described in 1982 in *Out of the Crisis*.[2] Tom Gilb[3] and Richard Zultner[4] also explored PDSA application to software development in later works.

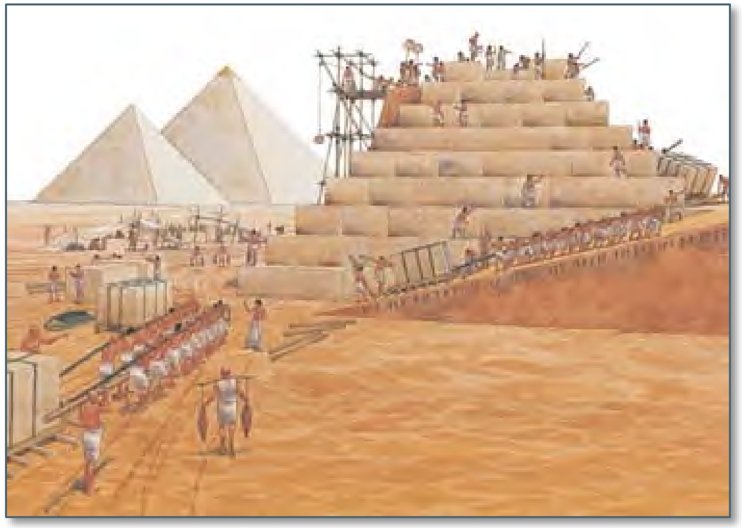
The X-15 hypersonic jet was a milestone 1950s project applying IID,[5] and the practice was considered a major contribution to the X-15's success. Although the X-15 was not a software project, it is noteworthy because some personnel—and hence, IID experience—seeded NASA's early 1960s Project Mercury, which did apply IID in software. In addition, some Project Mercury personnel seeded the IBM Federal Systems Division (FSD), another early IID proponent.

Project Mercury ran with very short (half-day) iterations that were time boxed. The development team conducted a technical review of all changes, and, interestingly, applied the Extreme Programming practice of test-first development, planning and writing tests before each micro-increment. They also practiced top-down development with stubs.

- 2.3 million blocks
- 6 million tons
- 140 meters high
- Tallest man-made structure for 3800 years

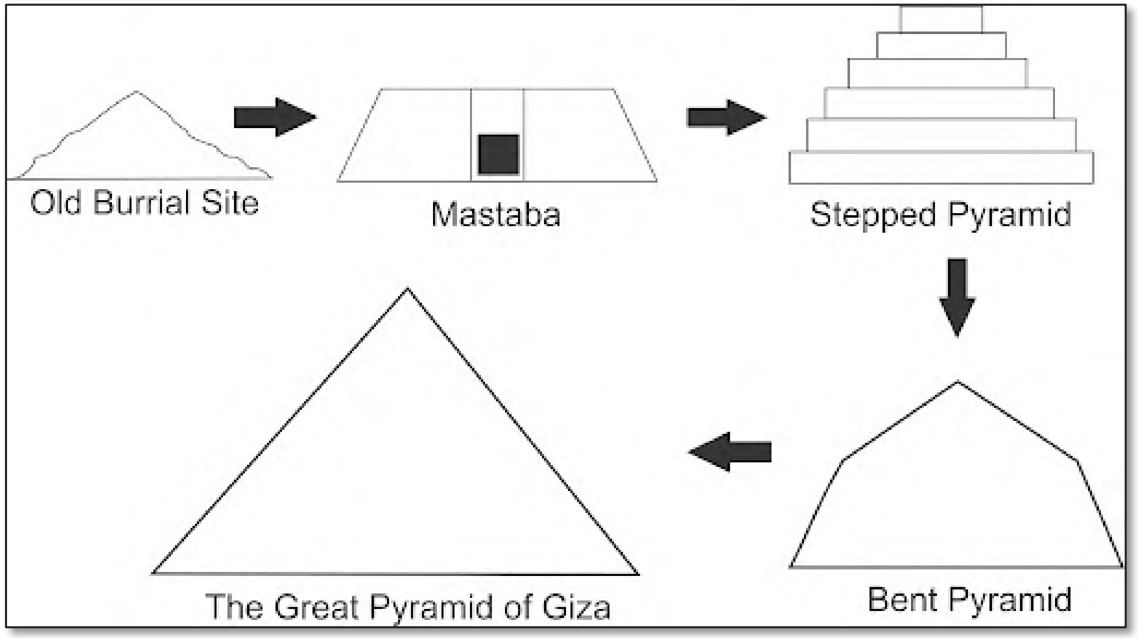
Henrik Kniberg

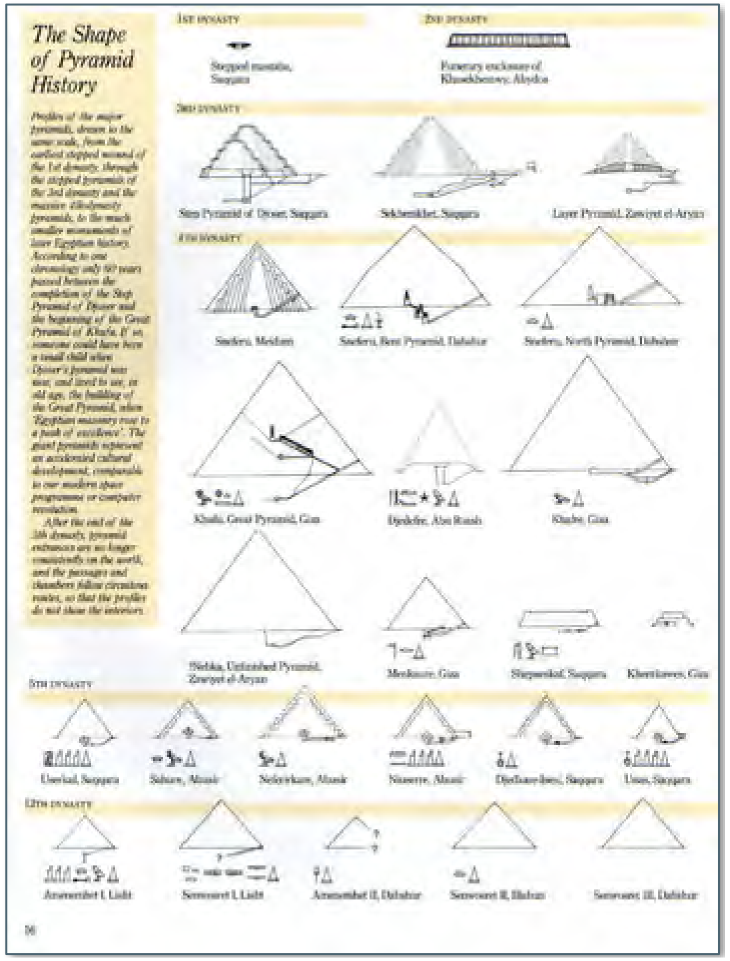
Iterations, Continuous Improvement, Pull, Single-piece flow 4500 years ago



Velocity of Khufu's pyramid construction:
1 block every 2.5 minutes
... for 30 years!!!

2.5 – 15 tons





The Shape of Pyramid History

Profiles of the major pyramids, drawn to the same scale, from the earliest stepped mound of the 1st dynasty, through the stepped pyramids of the 3rd dynasty and the massive 4th-dynasty pyramids, to the much smaller monuments of later Egyptian history. According to one chronology only 60 years passed between the completion of the Step Pyramid of Djoser and the beginning of the Great Pyramid of Khufu. If so, someone could have been a small child when Djoser's pyramid was new, and lived to see, in old age, the building of the Great Pyramid, when 'Egyptian masonry rose to a peak of excellence'. The giant pyramids represent an accelerated cultural development, comparable to our modern space programme or computer revolution.

After the end of the 5th dynasty, pyramid entrances are no longer consistently on the north, and the passages and chambers follow circuitous routes, so that the profiles do not show the interiors.

Henrik Kniberg

Beware of Tool Misuse

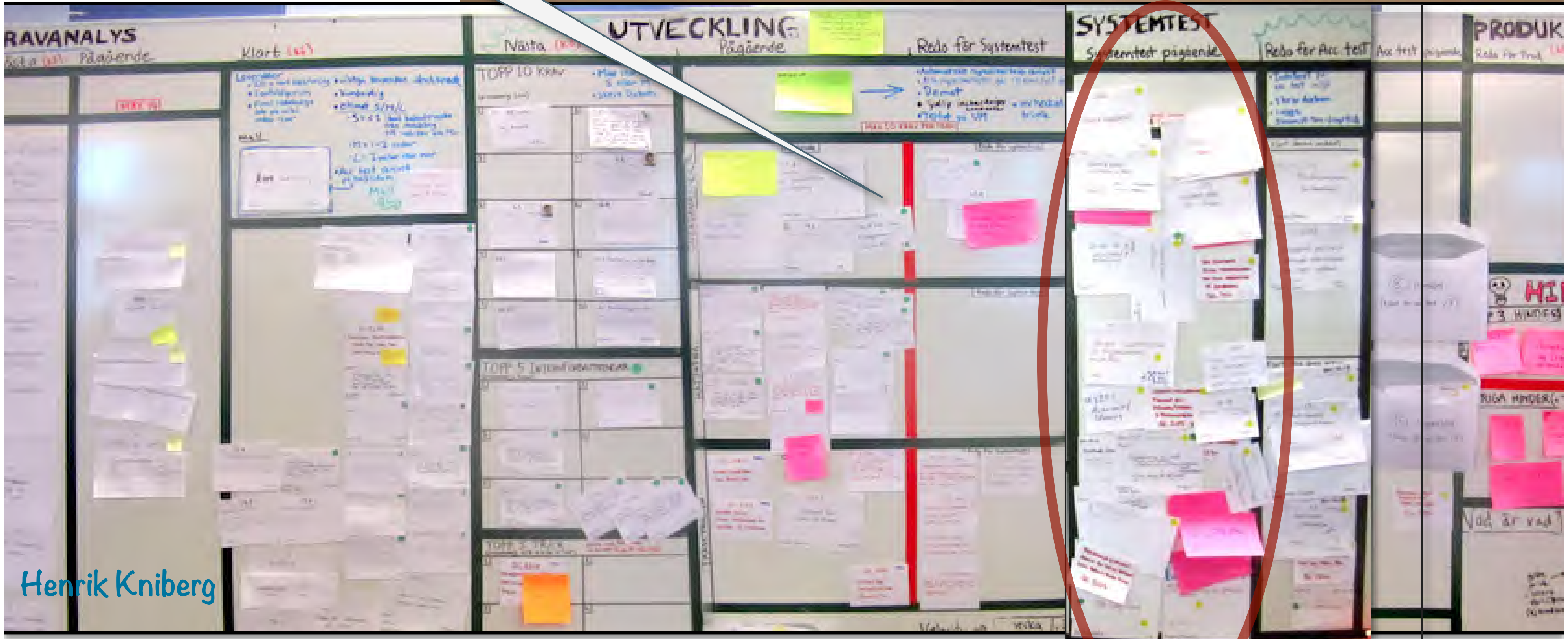
If all you have is a hammer, everything looks like a nail

Abraham Maslow

Henrik Kniberg

Misguided Lean

Photo: http://leanactionplan.pl/o-nas/artykuly_lean/Lean-Office;183.html

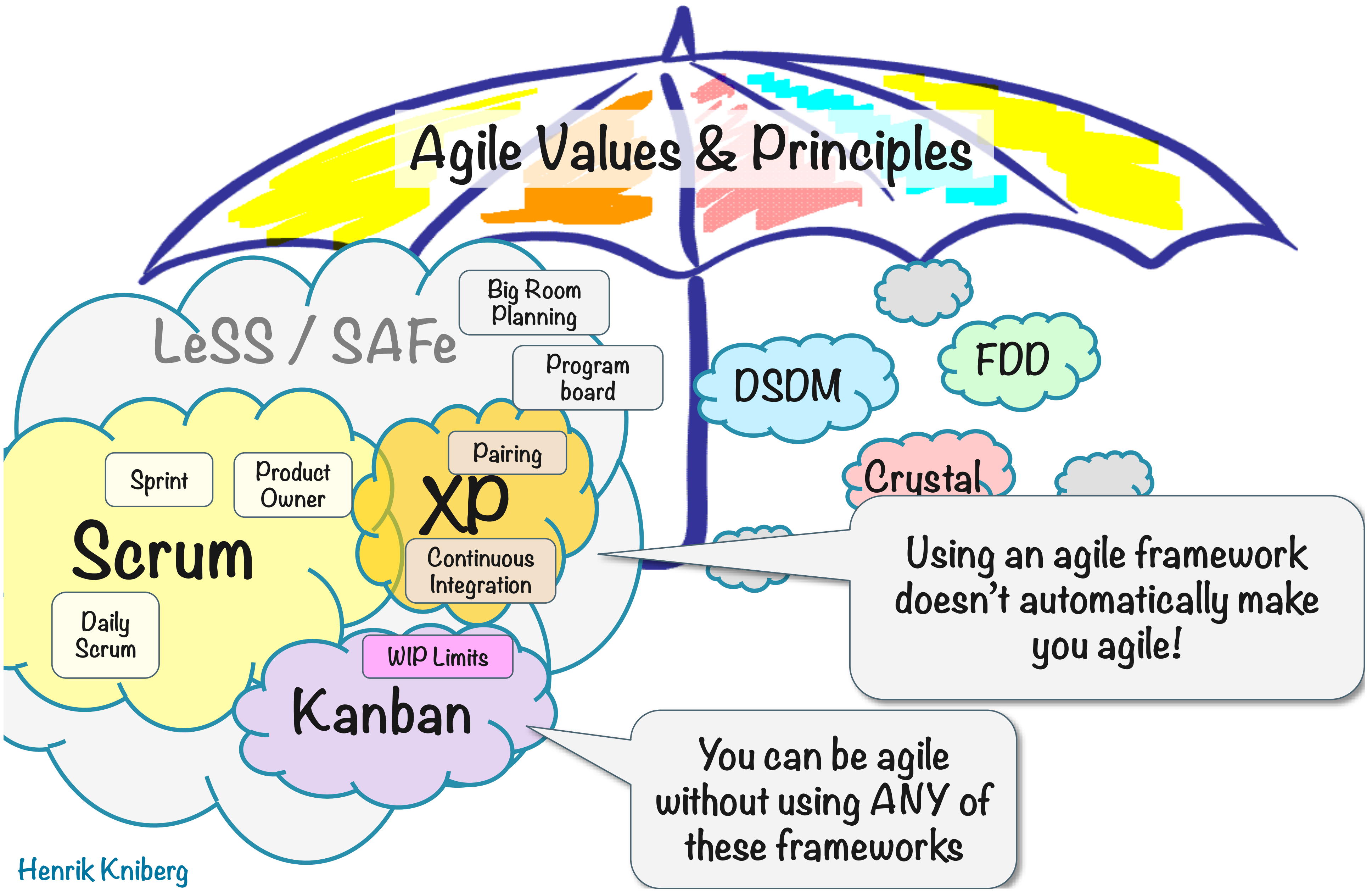


Solving the wrong problem

Revealing the right problem

Henrik Kniberg

The Agile "umbrella"



Henrik Kniberg

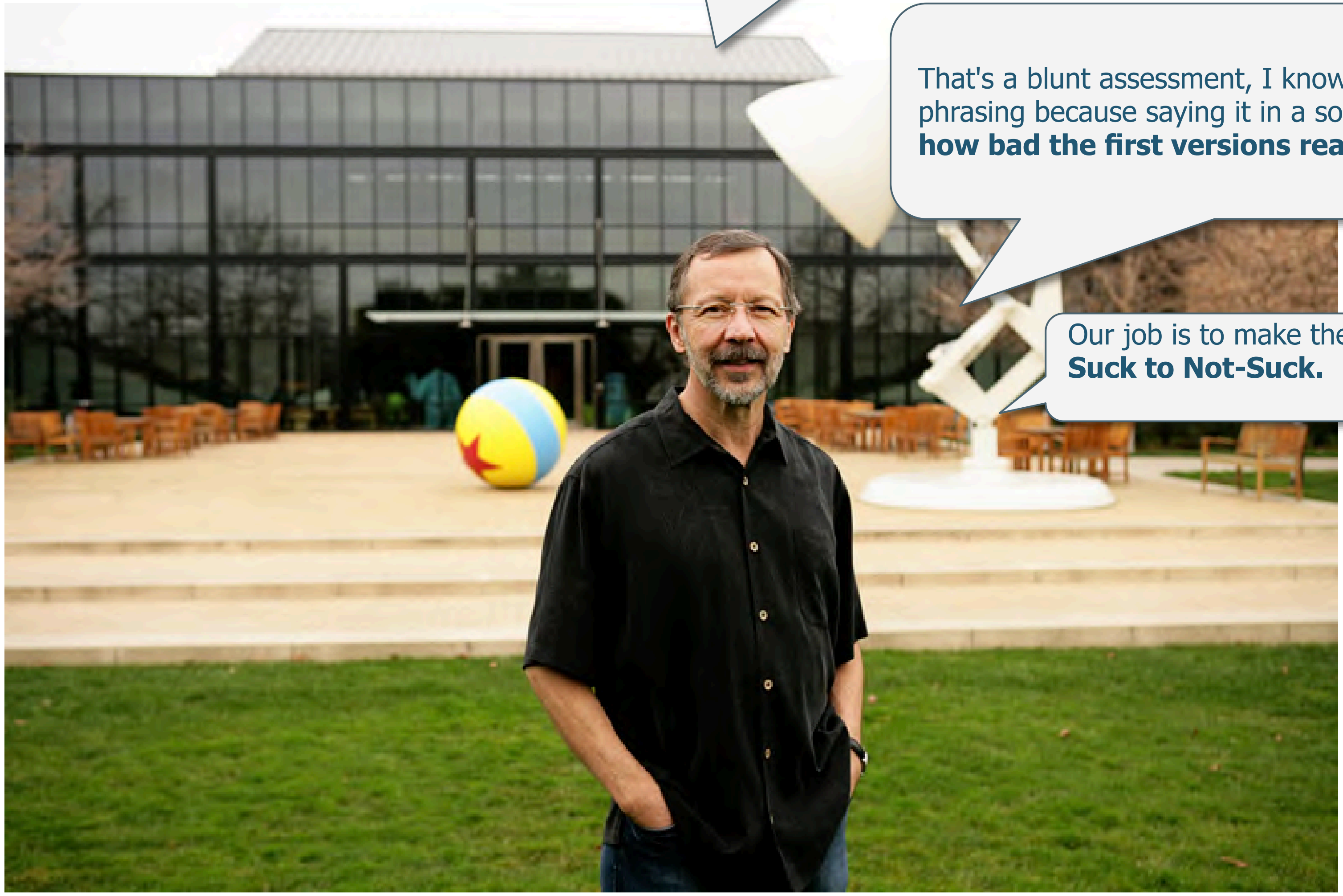
Henrik Kniberg

Example: Pixar

Early on, **all of our movies suck.**

That's a blunt assessment, I know, but I choose that phrasing because saying it in a softer way fails to convey **how bad the first versions really are.**

Our job is to make them go from **Suck to Not-Suck.**

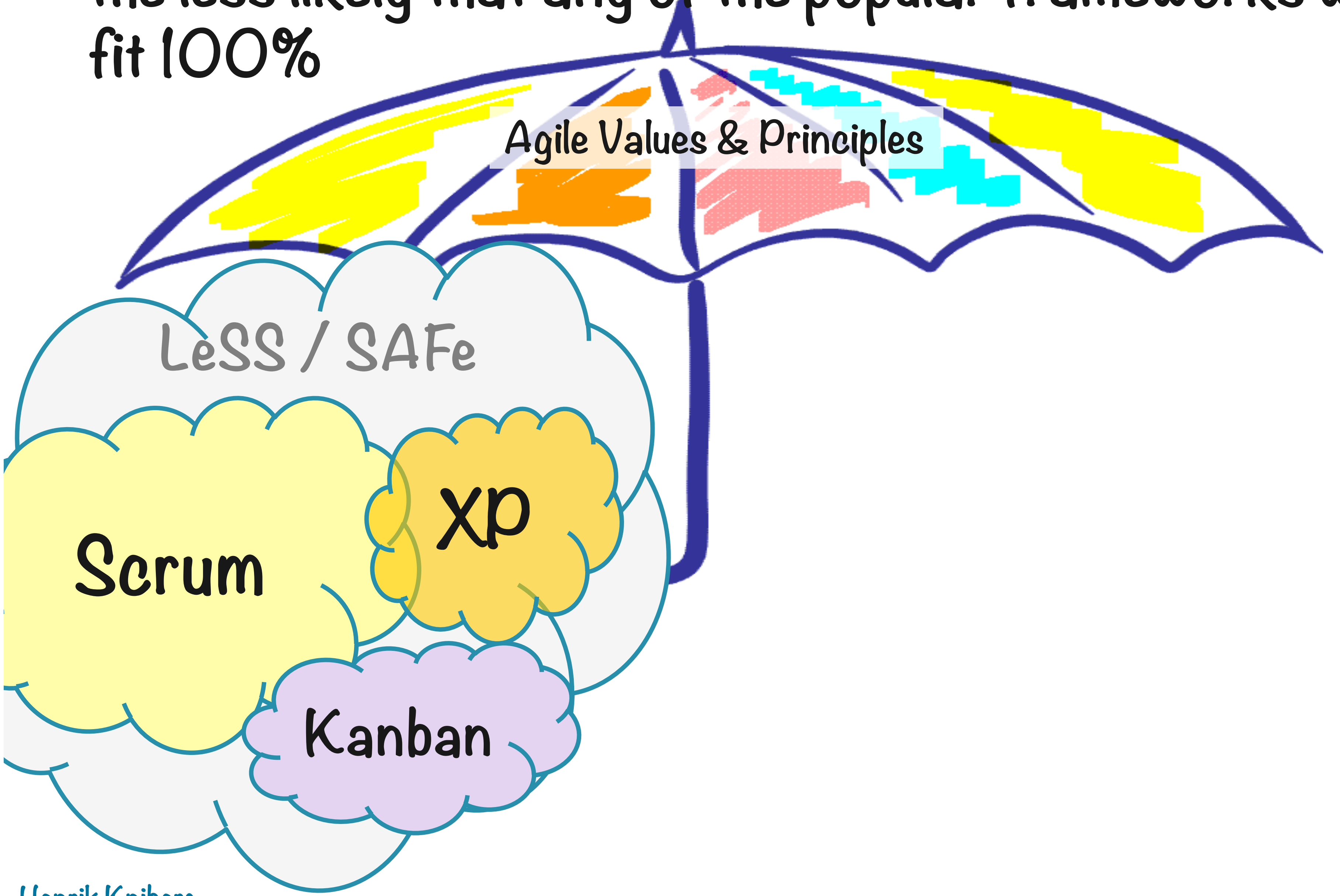


Ed Catmull
President of Pixar & Disney Animation Studios

In the early stage of making a movie, we draw storyboards (a comic-book version of the story) and then edit them together with dialogue and temporary music.
The first versions are very rough, but they give a sense of what the problems are, which in the beginning of all productions are many.
We then iterate, and each version typically gets better and better.

Henrik Kniberg

The further you are from software development, the less likely that any of the popular frameworks will fit 100%



Henrik Kniberg

Understand the Why of each tool

Story points

Retrospective

Definition of Done

Cross-functional team

Scrum Master

User stories

XP

Daily standup

Continuous Integration

Cadence

WIP limits

Sprint

Velocity

Value stream mapping

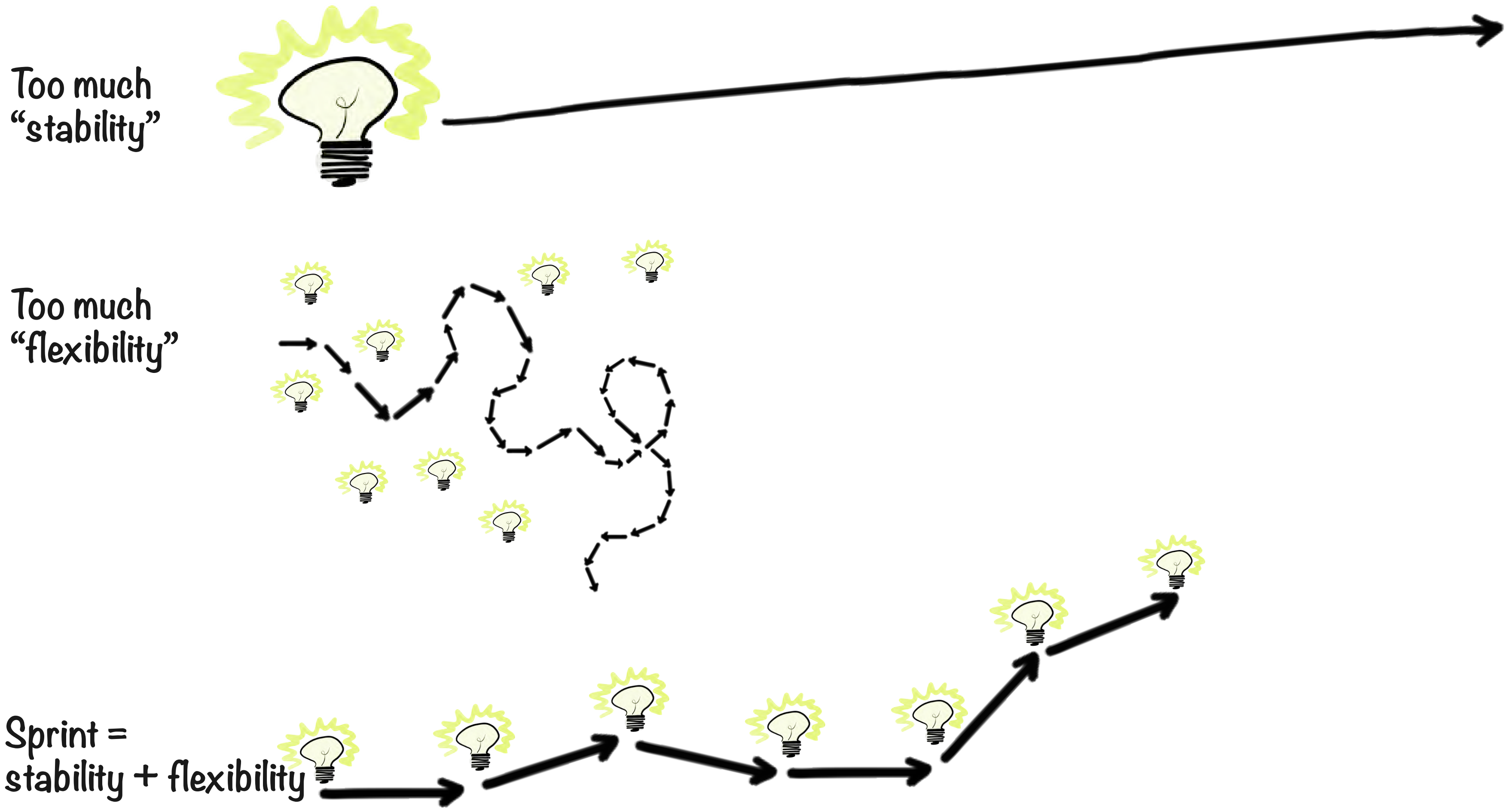
TDD

Pair programming

Henrik Kniberg

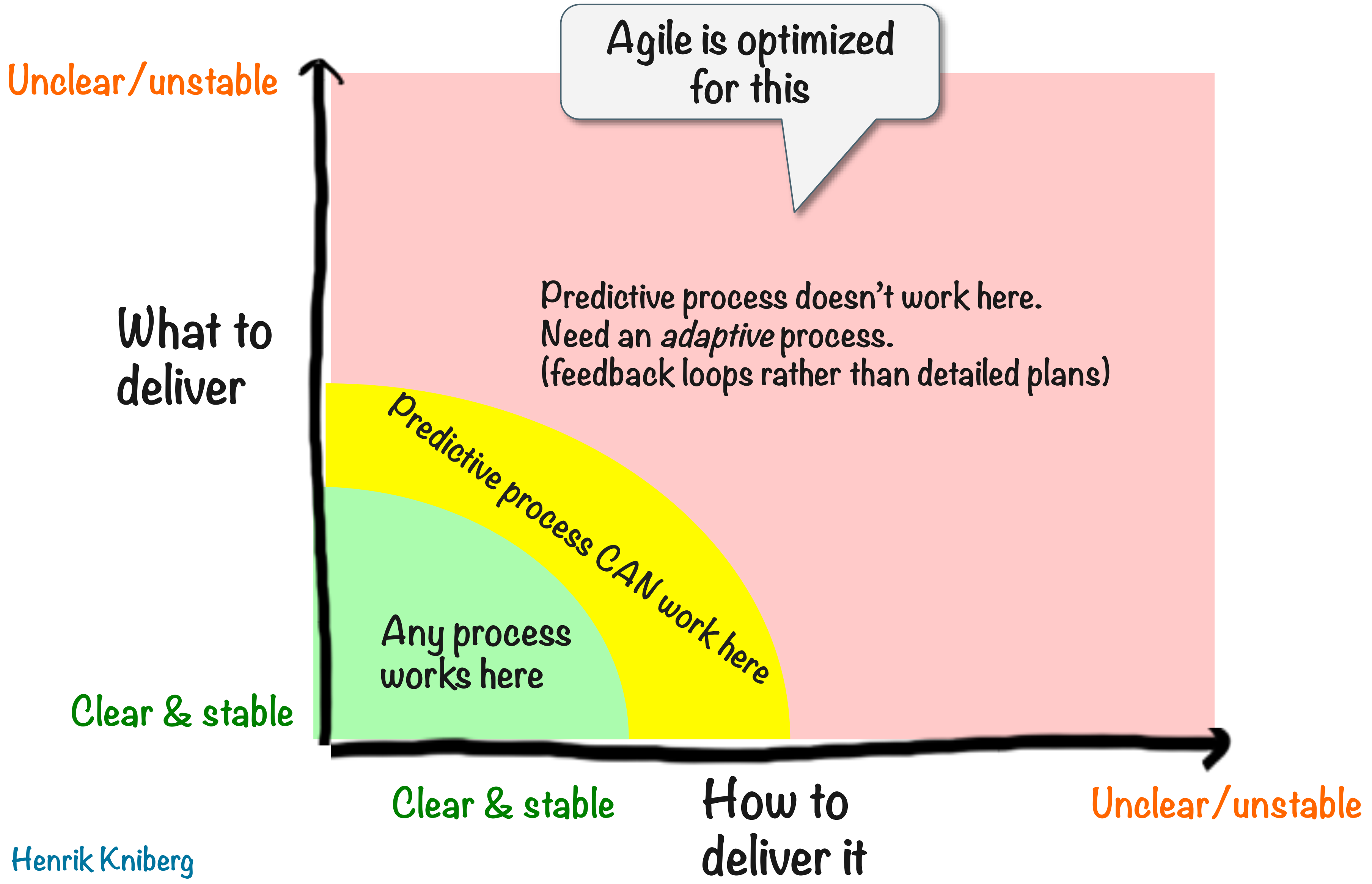
Example: Why Sprints?

Compromise between stability & flexibility.



Henrik Kniberg

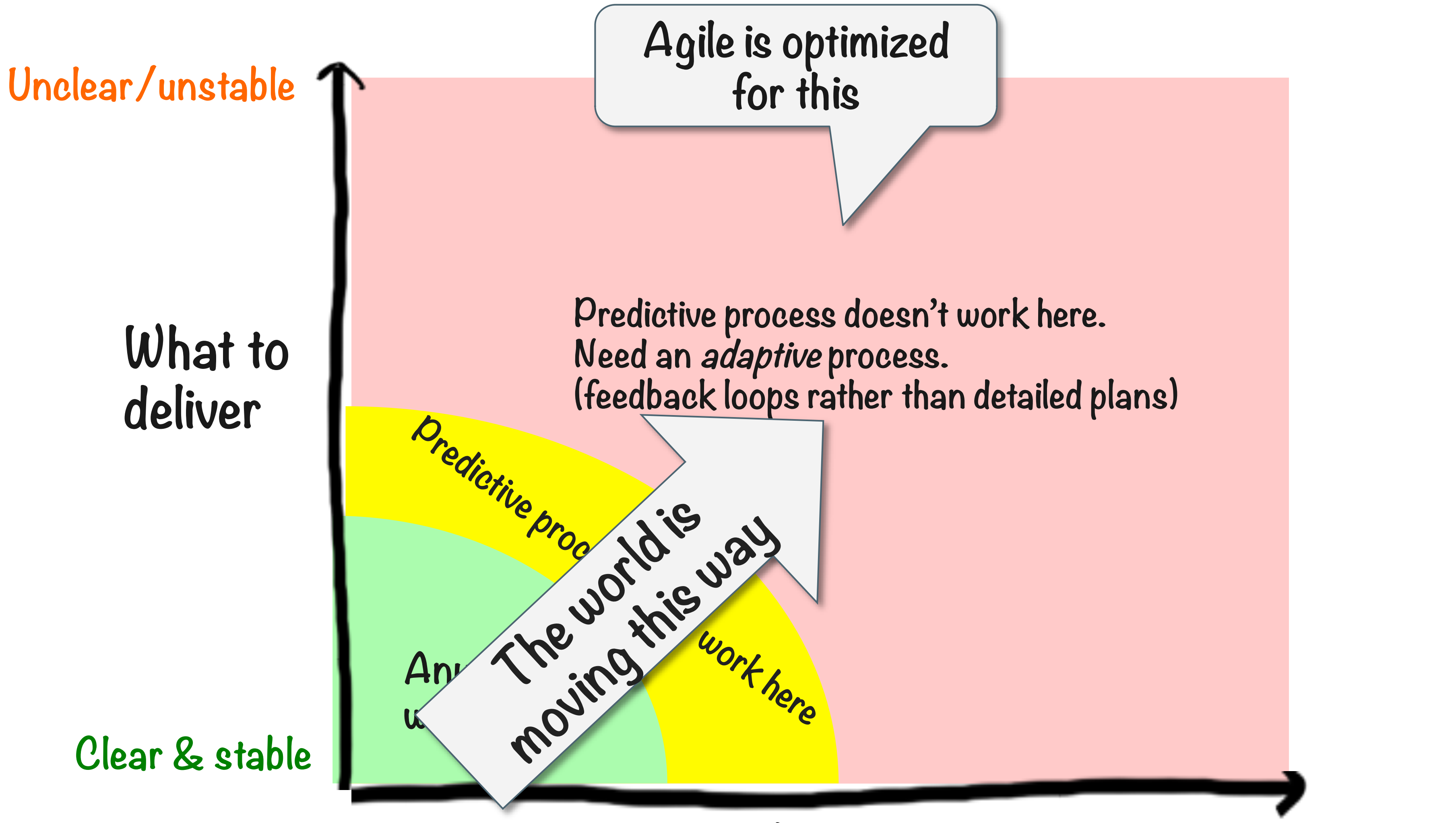
When is Agile most needed?



Agile is optimized for this
Unclear/unstable
What to deliver
Clear & stable
Predictive process doesn't work here.
Need an *adaptive* process.
(feedback loops rather than detailed plans)
Predictive process CAN work here
Any process works here
Clear & stable
How to deliver it
Unclear/unstable

Henrik Kniberg

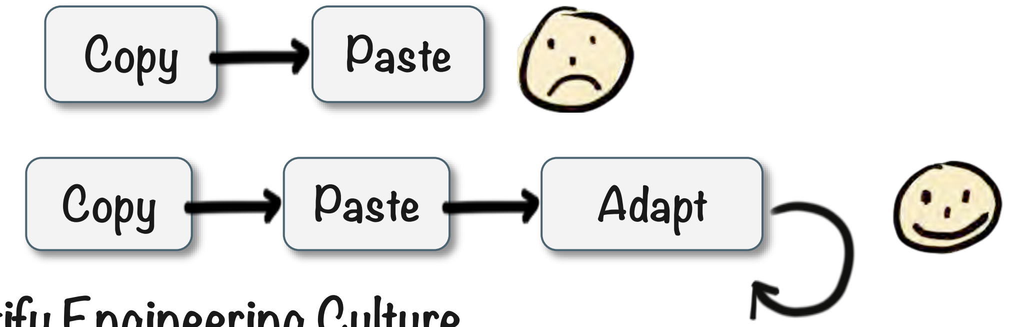
Why is Agile spreading so fast?



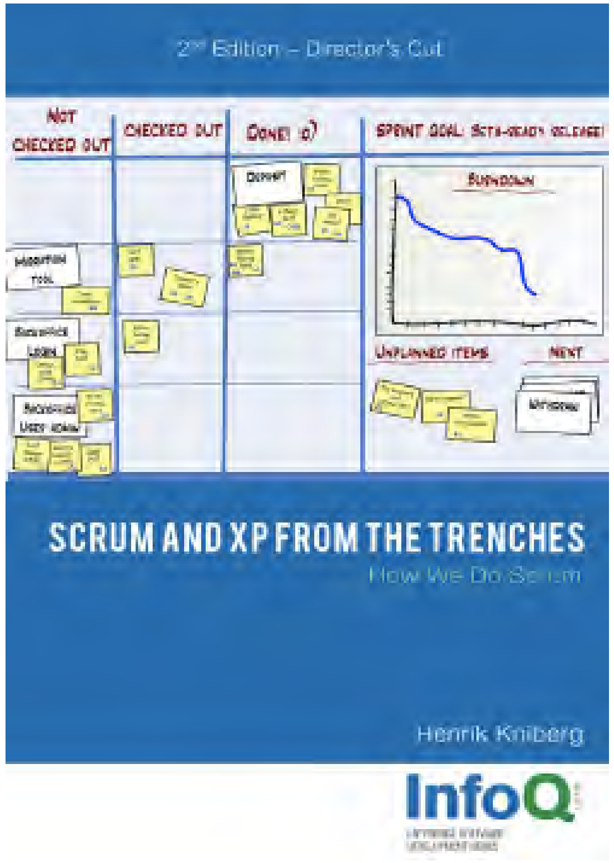
Agile is optimized for this
Unclear/unstable
What to deliver
Clear & stable
Predictive process doesn't work here.
Need an *adaptive* process.
(feedback loops rather than detailed plans)
Predictive proc
work here
An
w
The world is moving this way
Clear & stable
How to deliver it
Unclear/unstable

Henrik Kniberg

The role of copy-paste



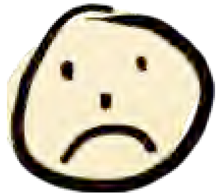
Scrum and XP from the Trenches



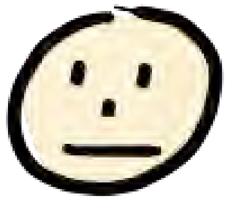
Spotify Engineering Culture

Strategies for applying agile in other contexts

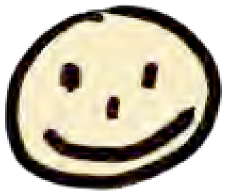
Implement method X "by the book", and follow the rules religously



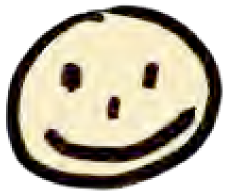
Implement method X "by the book", then customize it



Cherry-pick specific practices



Apply agile ideas directly, without using any specific framework



Henrik Kniberg

Strategies for applying agile in other contexts

Implement method X "by the book", and follow the rules religously

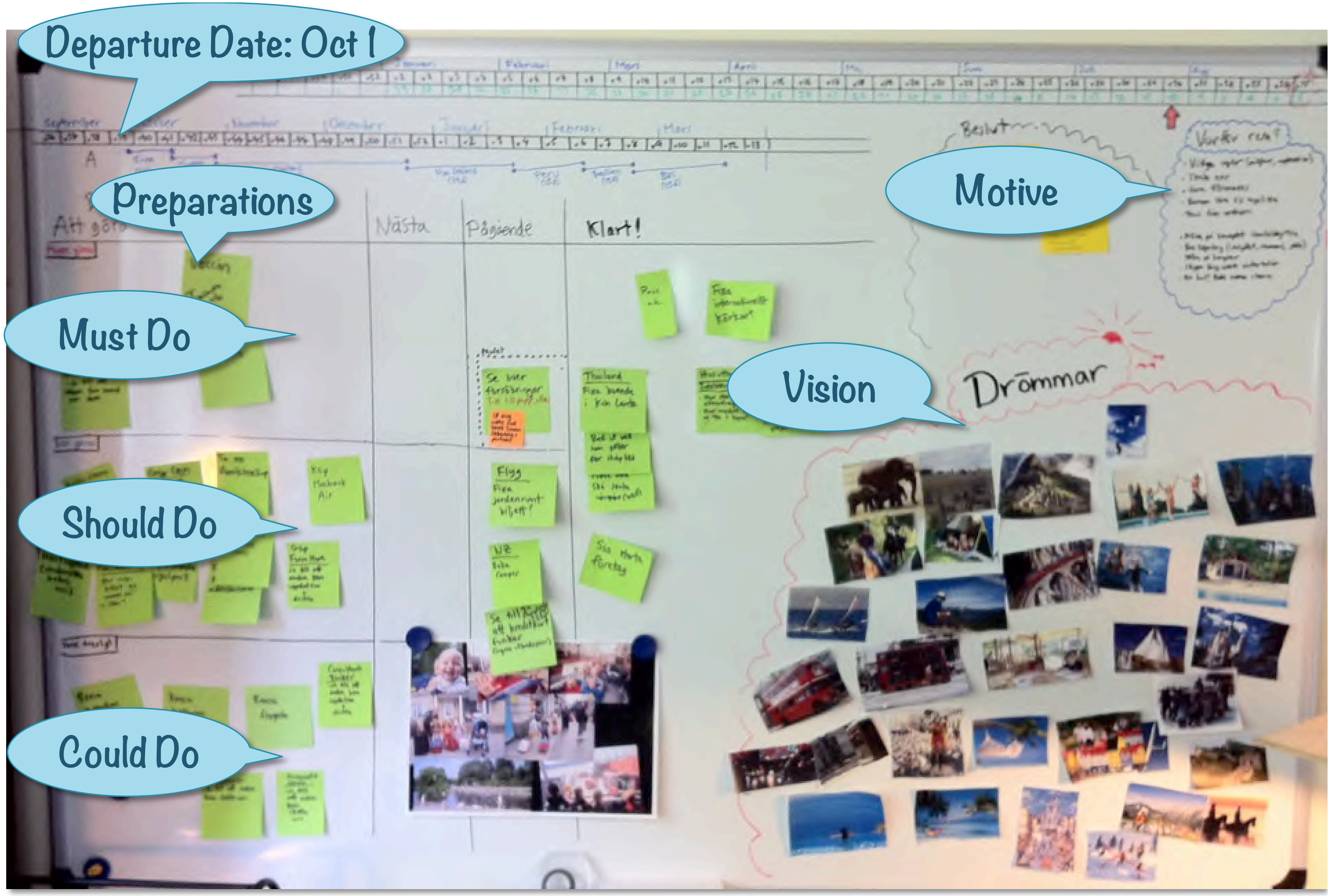
Implement method X "by the book", then customize it

Cherry-pick specific practices

Apply agile ideas directly, without using any specific framework

Henrik Kniberg

Example: Big Family Trip

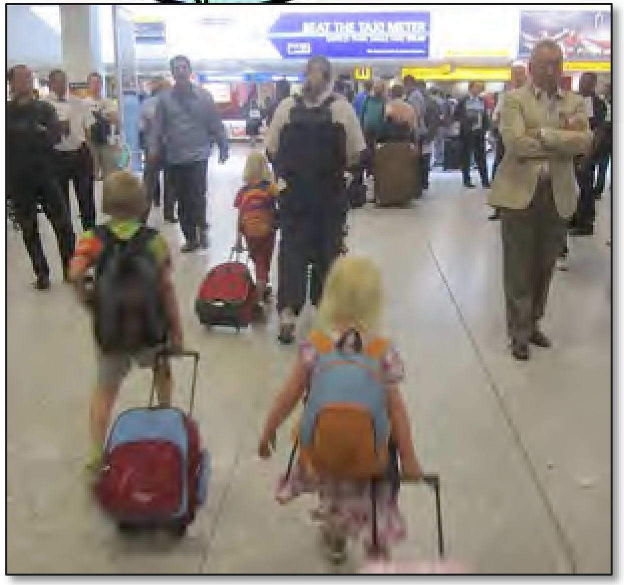


Henrik Kniberg

Travel "spike"

Small Family Trip

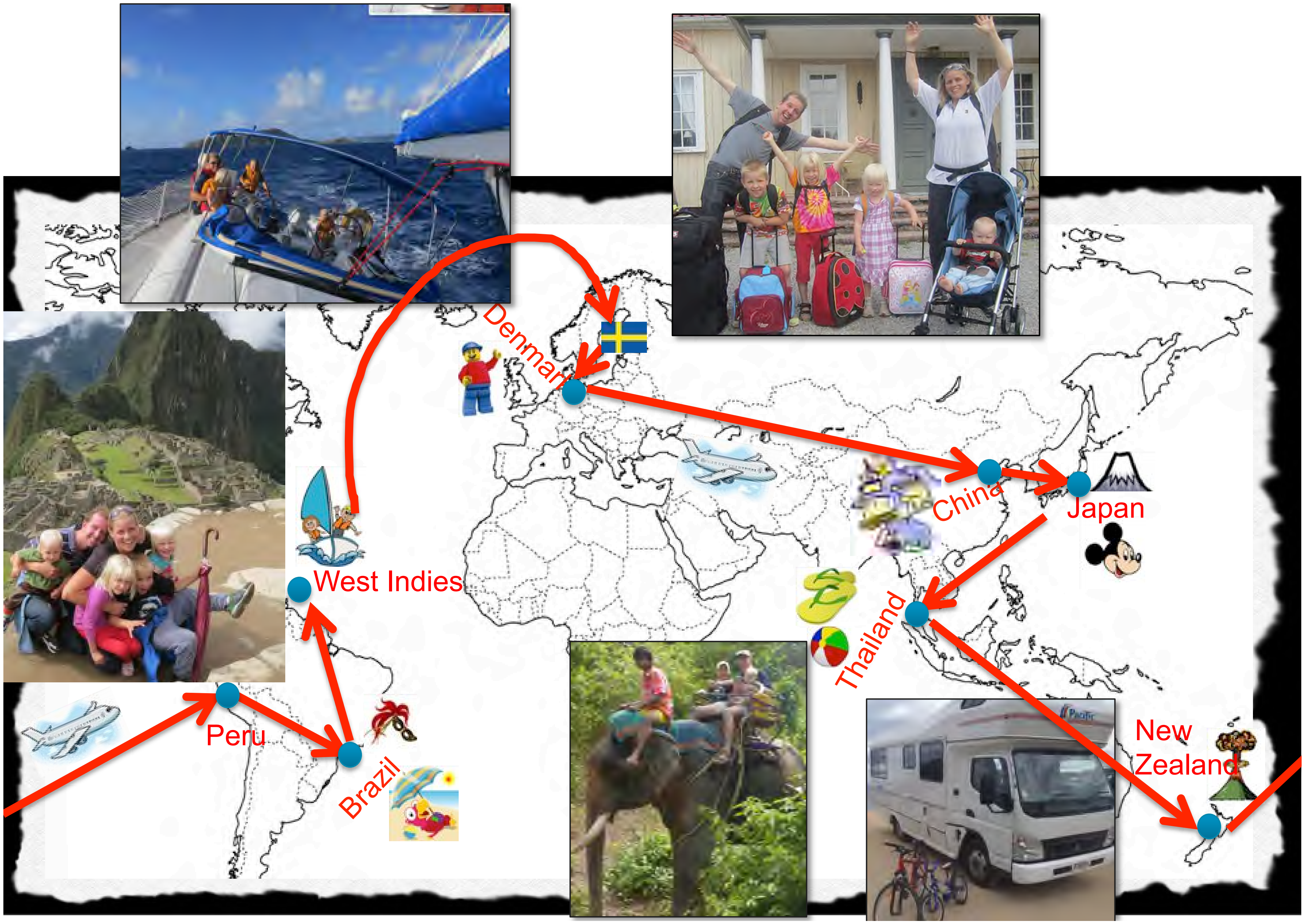
London, 4 days

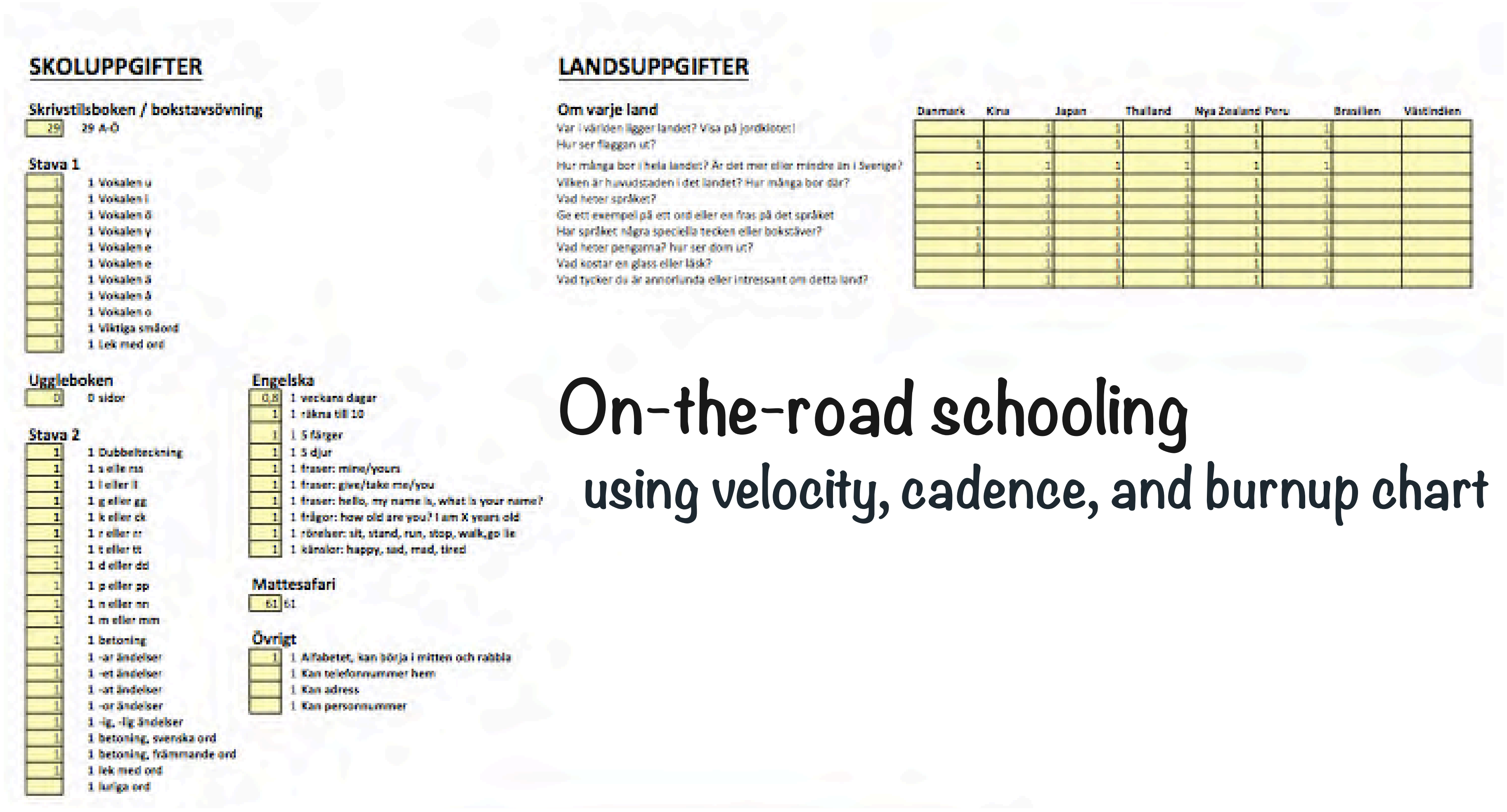


Big Family Trip

Round the world, 6 months

Henrik Kniberg


Denmark
China
Japan
West Indies
Thailand
Peru
Brazil
New Zealand



SKOLUPPGIFTER

Skrivstilsboken / bokstavsövning

29	29 A-Ö

Stava 1

1	1 Vokalen u
1	1 Vokalen i
1	1 Vokalen ö
1	1 Vokalen y
1	1 Vokalen e
1	1 Vokalen e
1	1 Vokalen å
1	1 Vokalen ä
1	1 Vokalen o
1	1 Viktiga småord
1	1 Lek med ord

Uggleboken

0	0 sidor

Stava 2

1	1 Dubbelteckning
1	1 s elle ss
1	1 l eller ll
1	1 g eller gg
1	1 k eller ck
1	1 r eller rr
1	1 t eller tt
1	1 d eller dd
1	1 p eller pp
1	1 n eller nn
1	1 m eller mm
1	1 betoning
1	1 -ar ändelser
1	1 -et ändelser
1	1 -at ändelser
1	1 -or ändelser
1	1 -ig, -lig ändelser
1	1 betoning, svenska ord
1	1 betoning, främmande ord
1	1 lek med ord
	1 lurliga ord

Engelska

0,8	1 veckans dagar
1	1 räkna till 10
1	1 5 färger
1	1 5 djur
1	1 fraser: mine/yours
1	1 fraser: give/take me/you
1	1 fraser: hello, my name is, what is your name?
1	1 frågor: how old are you? I am X years old
1	1 rörelser: sit, stand, run, stop, walk,go lie
1	1 känslor: happy, sad, mad, tired

Mattesafari

61	61

Övrigt

1	1 Alfabetet, kan börja i mitten och rabbla
	1 Kan telefonnummer hem
	1 Kan adress
	1 Kan personnummer

LANDSUPPGIFTER

Om varje land	Danmark	Kina	Japan	Thailand	Nya Zealand	Peru	Brasilien	Västindien
Var i världen ligger landet? Visa på jordklotet!		1	1	1	1	1		
Hur ser flaggan ut?	1	1	1	1	1	1		
Hur många bor i hela landet? Är det mer eller mindre än i Sverige?	1	1	1	1	1	1		
Vilken är huvudstaden i det landet? Hur många bor där?		1	1	1	1	1		
Vad heter språket?	1	1	1	1	1	1		
Ge ett exempel på ett ord eller en fras på det språket		1	1	1	1	1		
Har språket några speciella tecken eller bokstäver?	1	1	1	1	1	1		
Vad heter pengarna? hur ser dom ut?	1	1	1	1	1	1		
Vad kostar en glass eller läsk?		1	1	1	1	1		
Vad tycker du är annorlunda eller intressant om detta land?		1	1	1	1	1		

On-the-road schooling

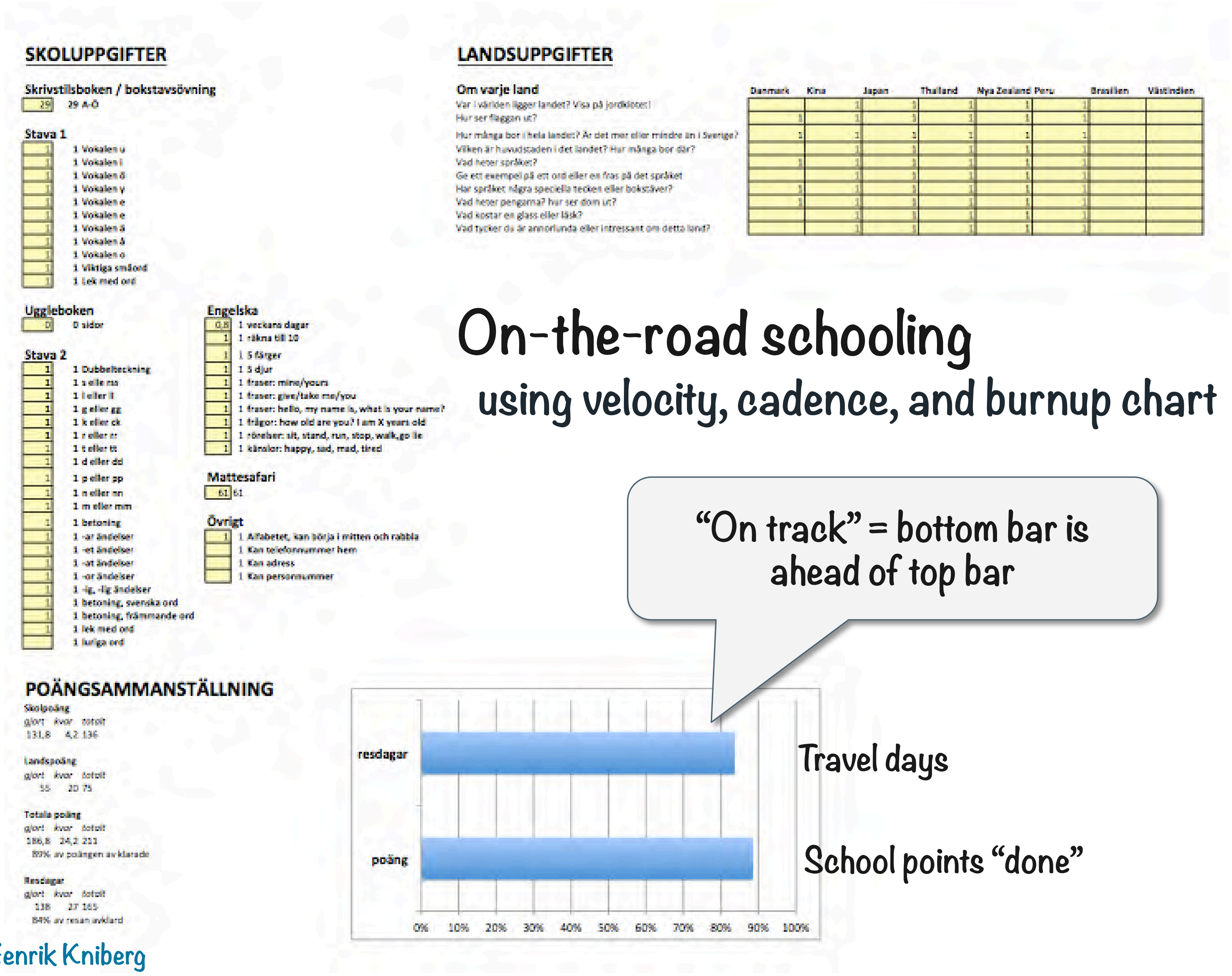
using velocity, cadence, and burnup chart

Henrik Kniberg

“School” is every day after breakfast, regardless of location

On-the-road schooling

using velocity, cadence, and burnup chart



SKOLUPPGIFTER

Skrivstilsboken / bokstavsövning

29	29 A-Ö

Stava 1

1	1 Vokalen u
1	1 Vokalen i
1	1 Vokalen ö
1	1 Vokalen y
1	1 Vokalen e
1	1 Vokalen e
1	1 Vokalen ä
1	1 Vokalen å
1	1 Vokalen o
1	1 Viktiga småord
1	1 Lek med ord

Uggleboken

0	0 sidor

Stava 2

1	1 Dubbelteckning
1	1 s elle ss
1	1 l eller ll
1	1 g eller gg
1	1 k eller ck
1	1 r eller rr
1	1 t eller tt
1	1 d eller dd
1	1 p eller pp
1	1 n eller nn
1	1 m eller mm
1	1 betoning
1	1 -ar ändelser
1	1 -et ändelser
1	1 -at ändelser
1	1 -or ändelser
1	1 -ig, -lig ändelser
1	1 betoning, svenska ord
1	1 betoning, främmande ord
1	1 lek med ord
	1 luriga ord

Engelska

0,8	1 veckans dagar
1	1 räkna till 10
1	1 5 färger
1	1 5 djur
1	1 fraser: mine/yours
1	1 fraser: give/take me/you
1	1 fraser: hello, my name is, what is your name?
1	1 frågor: how old are you? I am X years old
1	1 rörelser: sit, stand, run, stop, walk,go lie
1	1 känslor: happy, sad, mad, tired

Mattesafari

61	61

Övrigt

1	1 Alfabetet, kan börja i mitten och rabbla
	1 Kan telefonnummer hem
	1 Kan adress
	1 Kan personnummer

LANDSUPPGIFTER

Om varje land	Danmark	Kina	Japan	Thailand	Nya Zealand	Peru	Brasilien	Västindien
Var i världen ligger landet? Visa på jordklotet!		1	1	1	1	1		
Hur ser flaggan ut?	1	1	1	1	1	1		
Hur många bor i hela landet? Är det mer eller mindre än i Sverige?	1	1	1	1	1	1		
Vilken är huvudstaden i det landet? Hur många bor där?		1	1	1	1	1		
Vad heter språket?	1	1	1	1	1	1		
Ge ett exempel på ett ord eller en fras på det språket		1	1	1	1	1		
Har språket några speciella tecken eller bokstäver?	1	1	1	1	1	1		
Vad heter pengarna? hur ser dom ut?	1	1	1	1	1	1		
Vad kostar en glass eller läsk?		1	1	1	1	1		
Vad tycker du är annorlunda eller intressant om detta land?		1	1	1	1	1		

POÄNGSAMMANSTÄLLNING

Skolpoäng

gjort	kvar	totalt
131,8	4,2	136

Landspoäng

gjort	kvar	totalt
55	20	75

Totala poäng

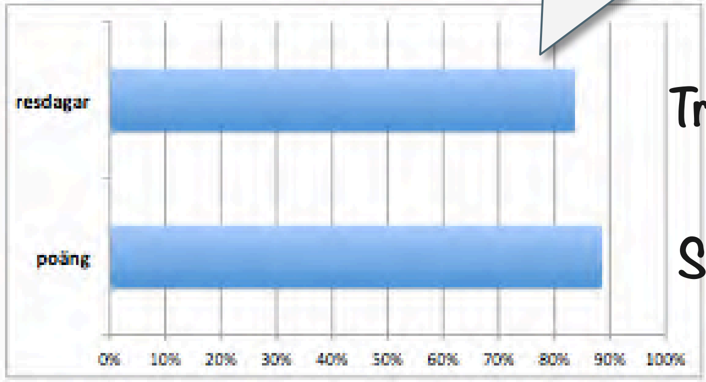
gjort	kvar	totalt
186,8	24,2	211

89% av poängen avklarade

Resdagar

gjort	kvar	totalt
138	27	165

84% av resan avklard



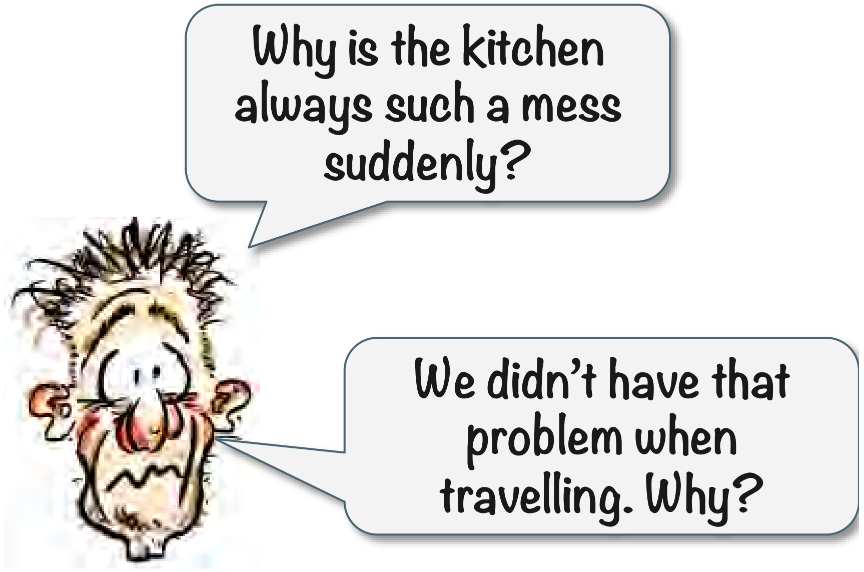
"On track" = bottom bar is ahead of top bar

Travel days

School points "done"

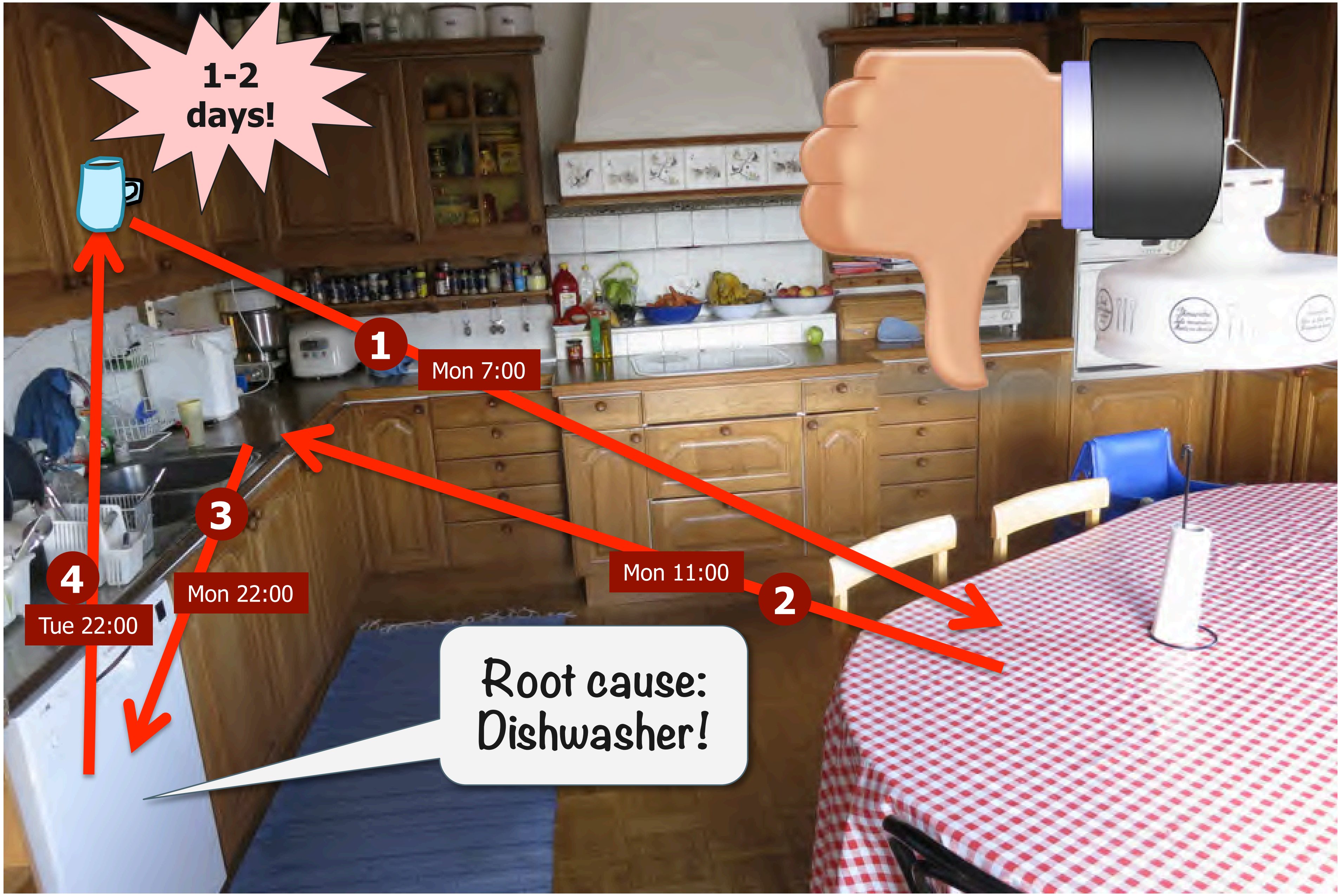
Henrik Kniberg

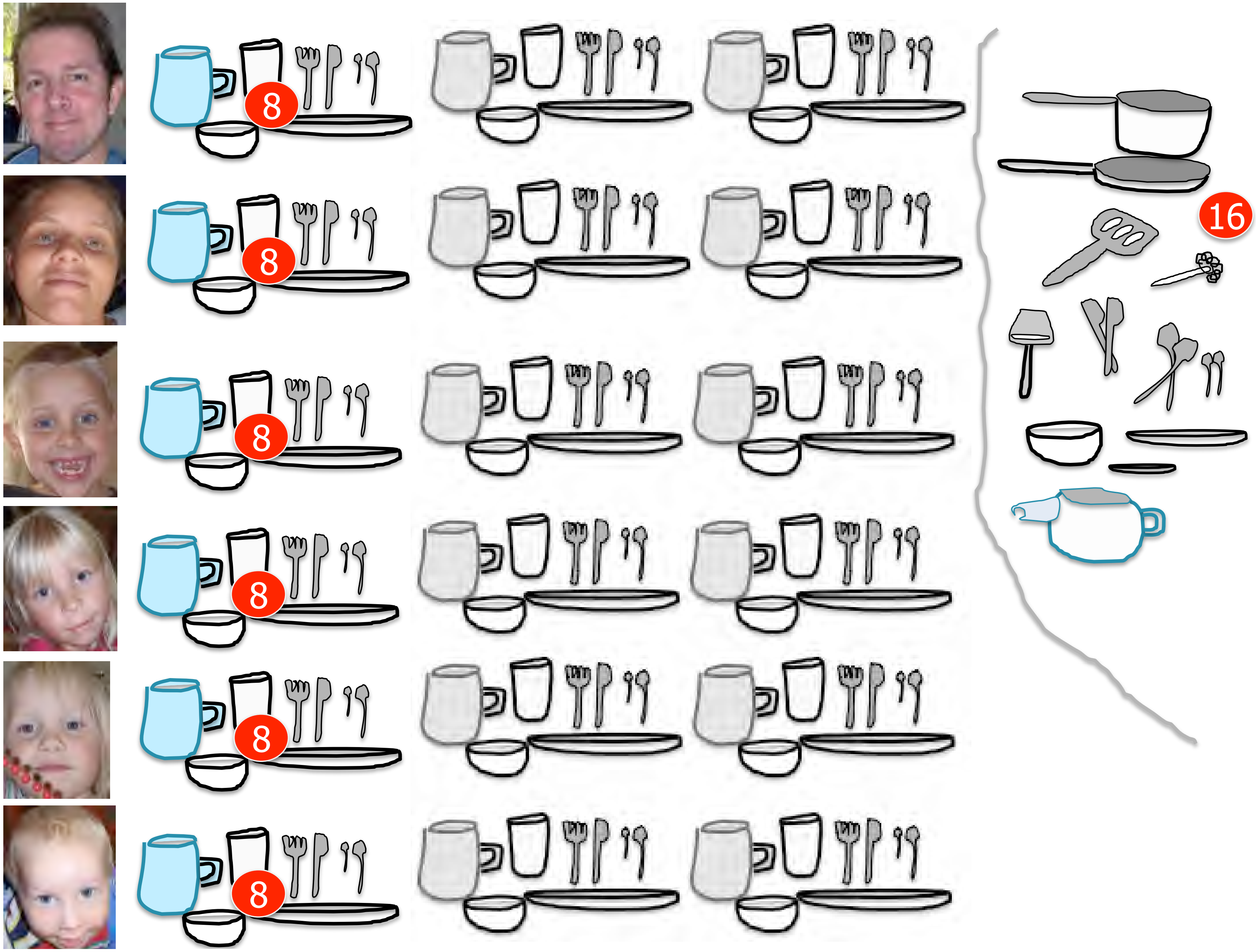
Back home from the trip...


Why is the kitchen always such a mess suddenly?
We didn't have that problem when travelling. Why?

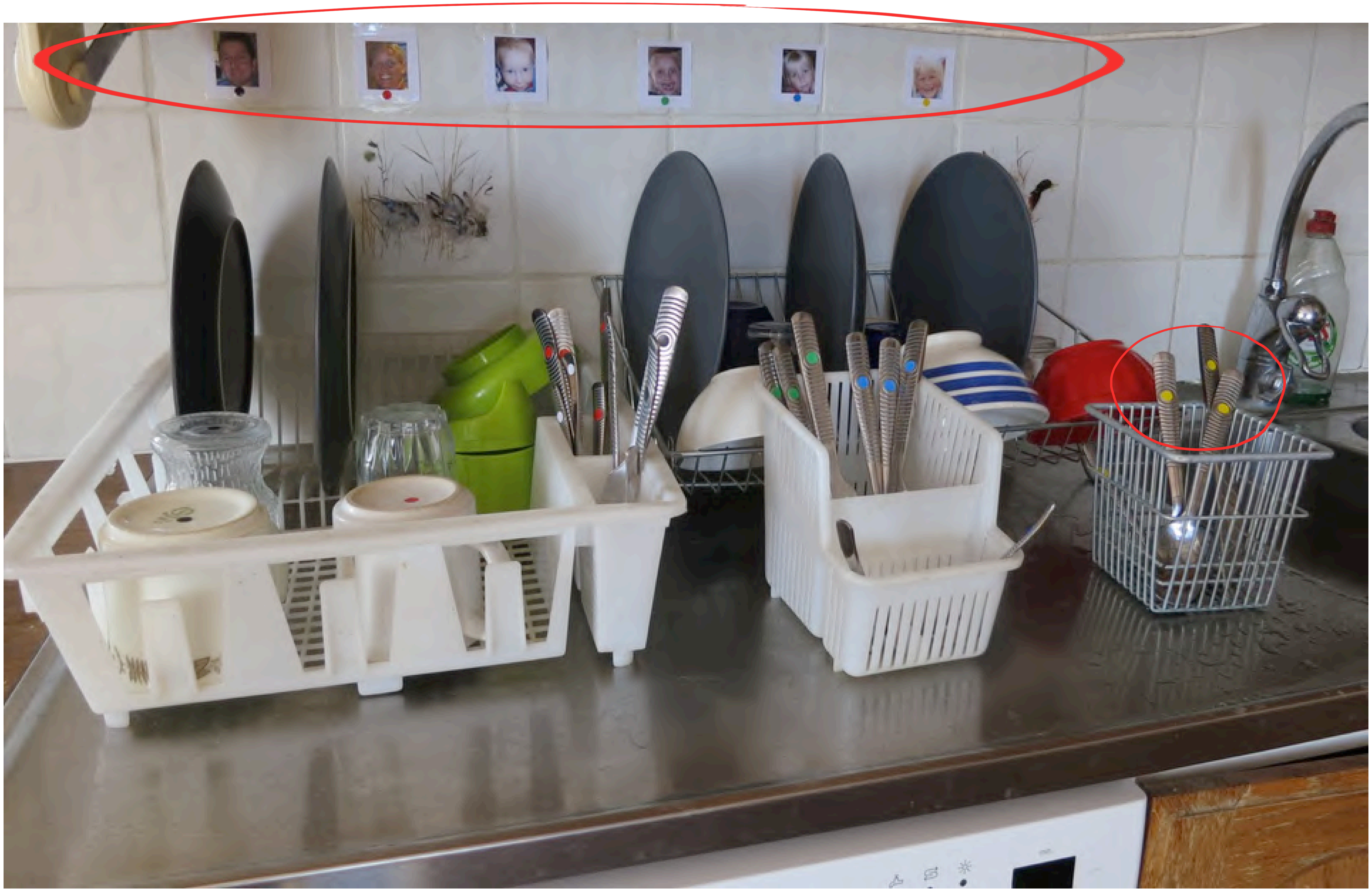
Henrik Kniberg

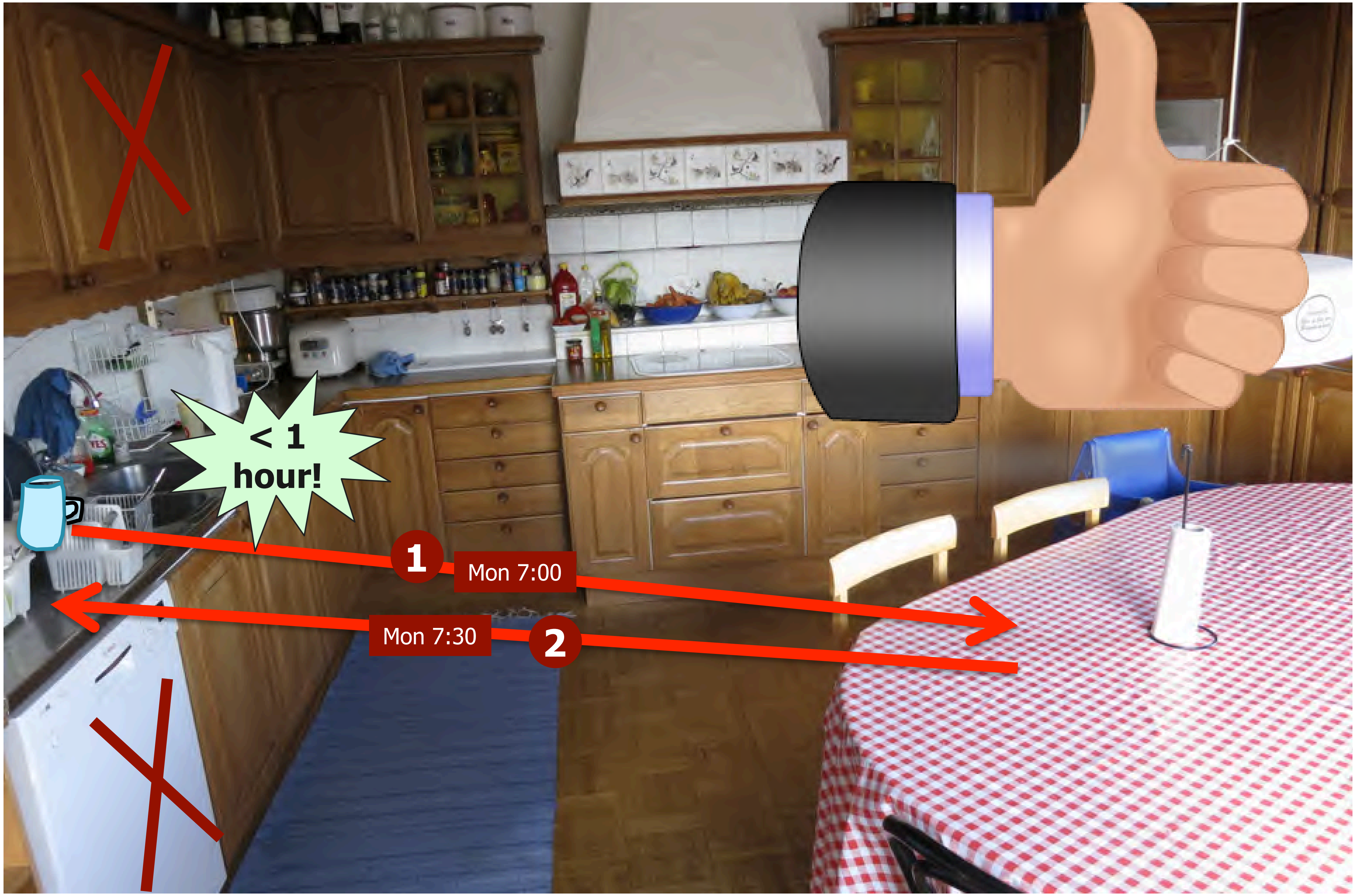
8
8
8
8
8
8
8
8
8
8
8
16
160+


1-2 days!
1
Mon 7:00
2
Mon 11:00
3
Mon 22:00
4
Tue 22:00
Root cause: Dishwasher!


8
8
8
8
8
8
16

Henrik Kniberg




< 1 hour!
1
Mon 7:00
Mon 7:30
2

Worked like a charm!
but did we keep doing it?

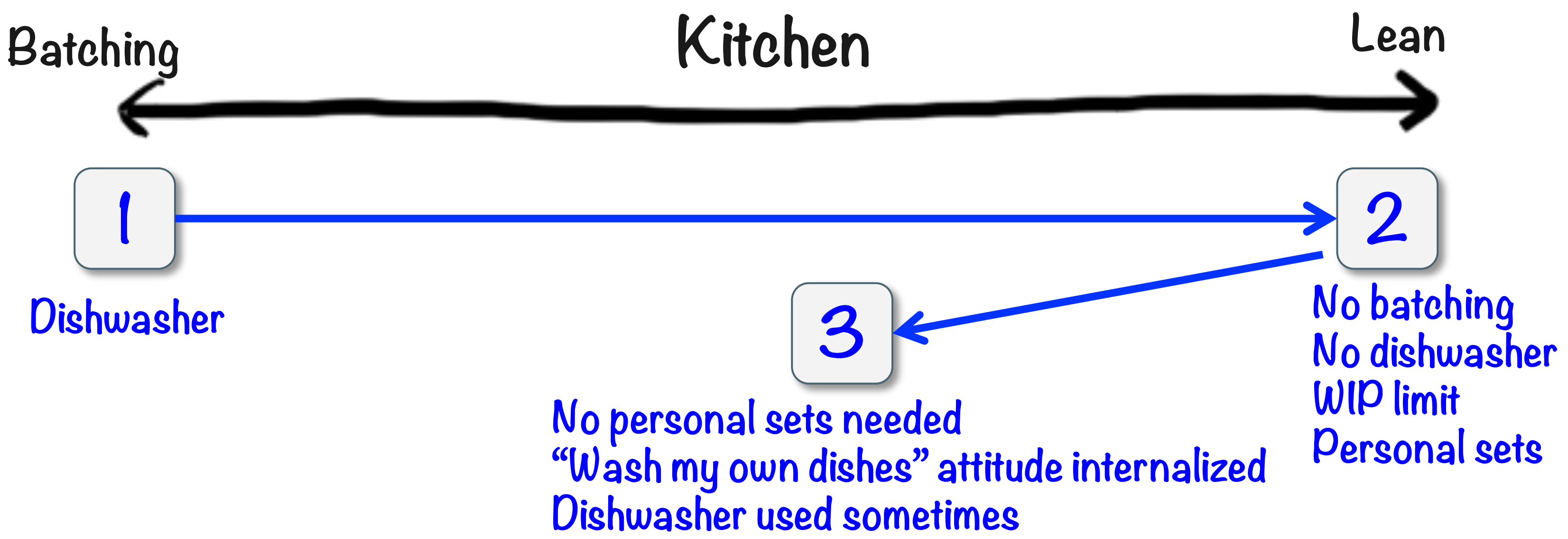
Henrik Kniberg

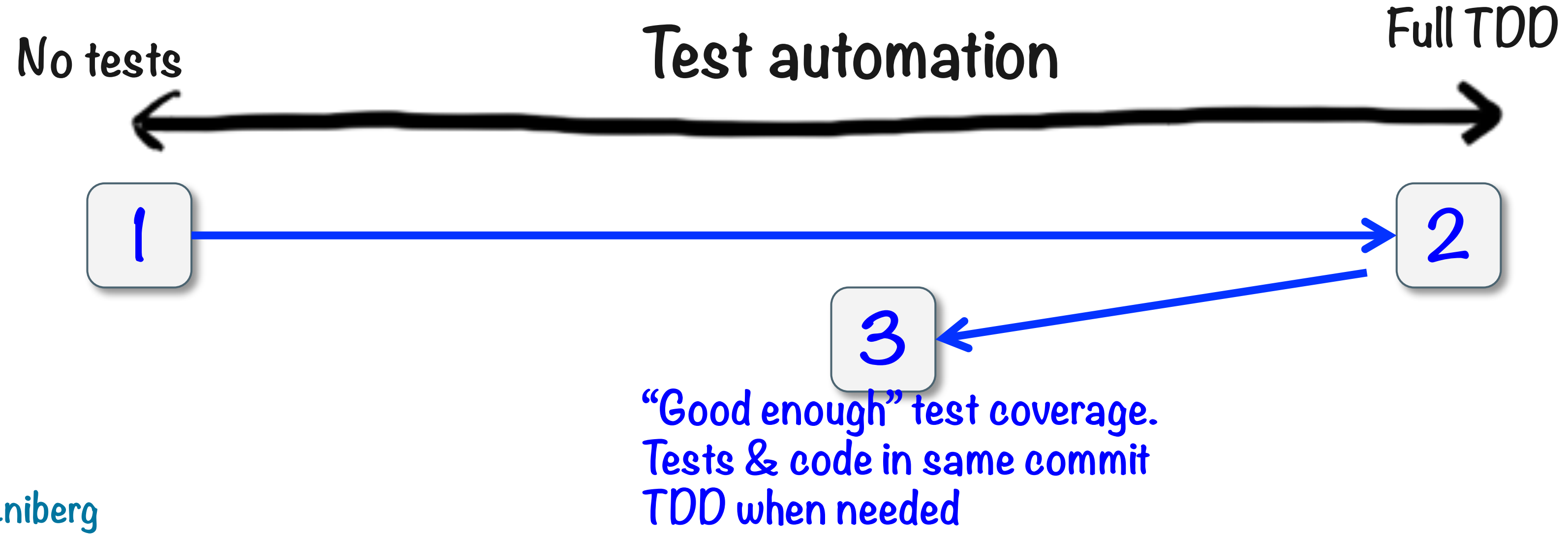
Sometimes agile practices don't stick.
That's Fine.

Explanations:

- The practice was only needed for a specific situation
- The practice didn't work too well
- The practice was a stepping stone until a better practice was found
- The practice was only needed to learn & internalize a new behaviour

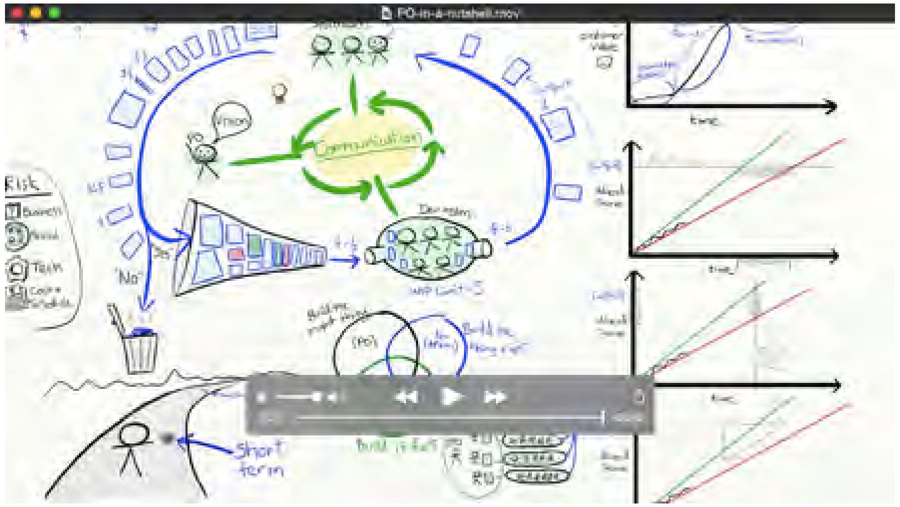
Pattern: Go all-in first, then go pragmatic





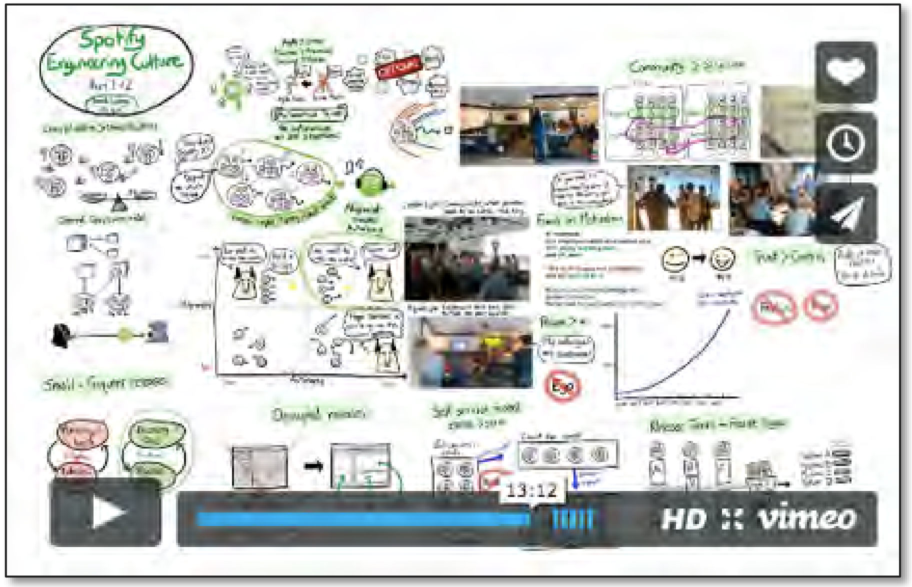
Henrik Kniberg

Example: Using a practice only when needed



Agile Product Ownership in a Nutshell
- Production time: 2 days

Takes a couple of days to make a cool animated video



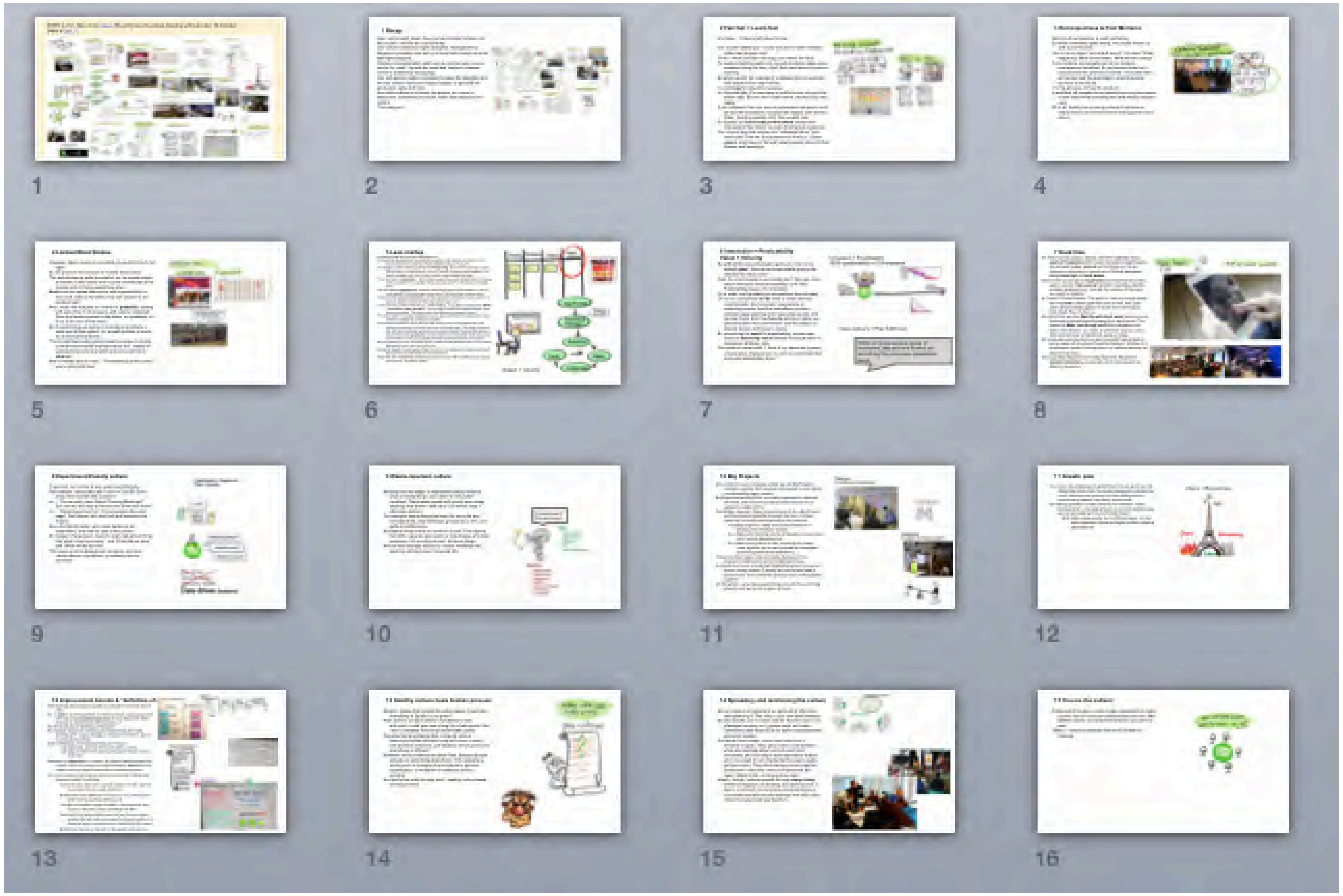
Spotify Engineering Culture video – part 1
- Expected production time: A few days
- Actual production time: Several weeks!

Whoa! That took MUCH longer than I expected!

How can I avoid the same problem for Part 2?

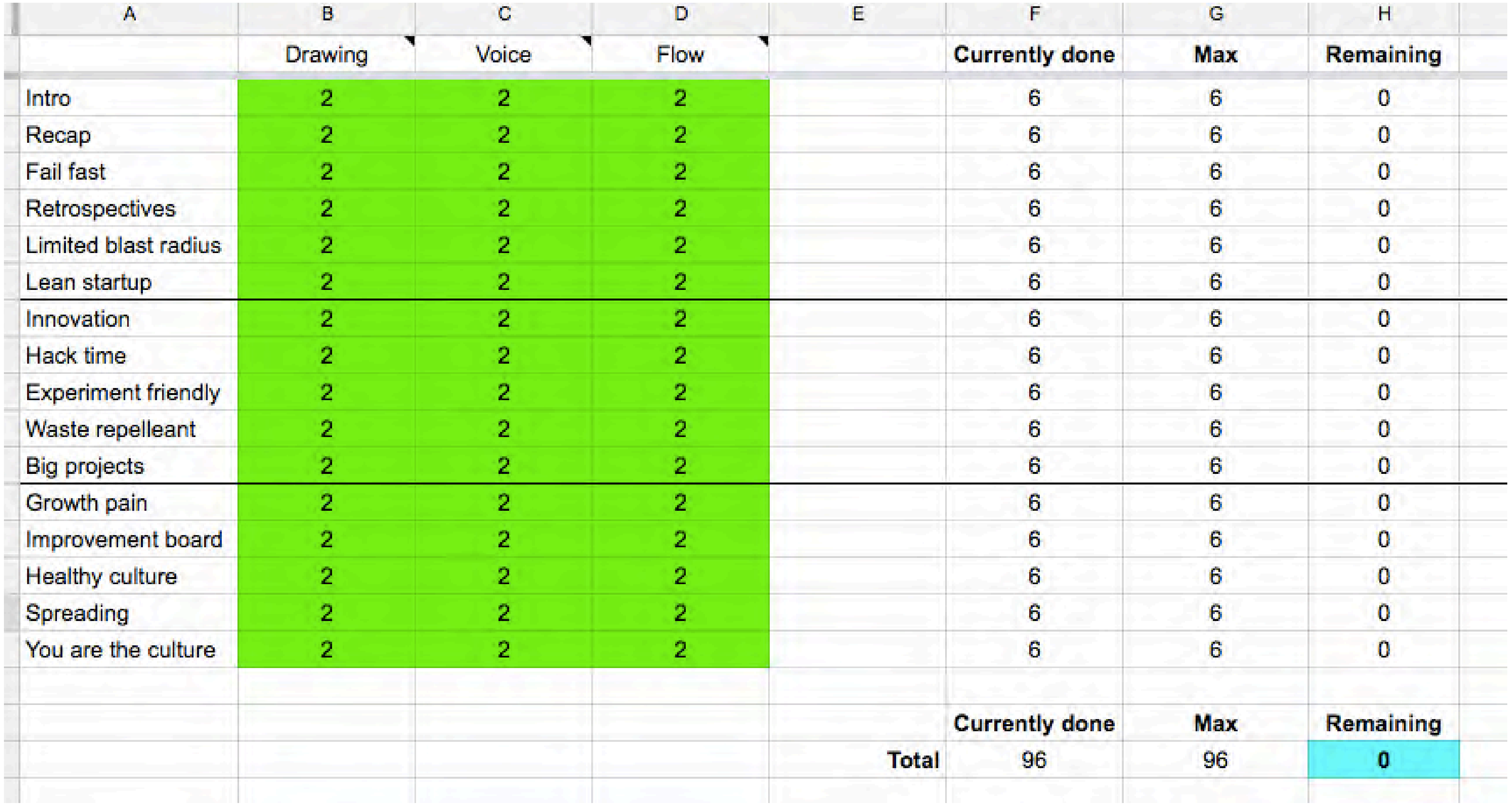
Henrik Kniberg

Video storyboard (rough sketches)



Henrik Kniberg

“Pointifying” the work

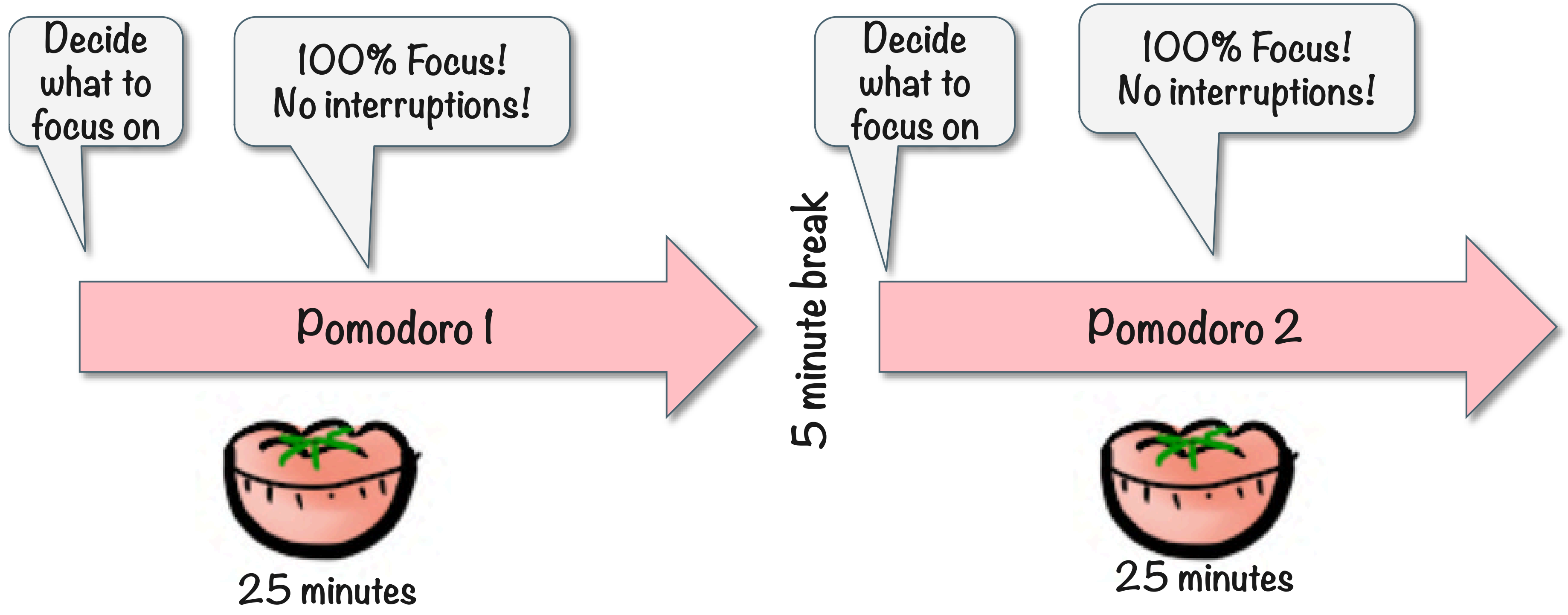


A	B	C	D	E	F	G	H
	Drawing	Voice	Flow		**Currently done**	**Max**	**Remaining**
Intro	2	2	2		6	6	0
Recap	2	2	2		6	6	0
Fail fast	2	2	2		6	6	0
Retrospectives	2	2	2		6	6	0
Limited blast radius	2	2	2		6	6	0
Lean startup	2	2	2		6	6	0
Innovation	2	2	2		6	6	0
Hack time	2	2	2		6	6	0
Experiment friendly	2	2	2		6	6	0
Waste repelleant	2	2	2		6	6	0
Big projects	2	2	2		6	6	0
Growth pain	2	2	2		6	6	0
Improvement board	2	2	2		6	6	0
Healthy culture	2	2	2		6	6	0
Spreading	2	2	2		6	6	0
You are the culture	2	2	2		6	6	0
					Currently done	**Max**	**Remaining**
				Total	96	96	**0**

Henrik Kniberg

Pomodoro Technique

Personal Scrum with 30 minute sprints

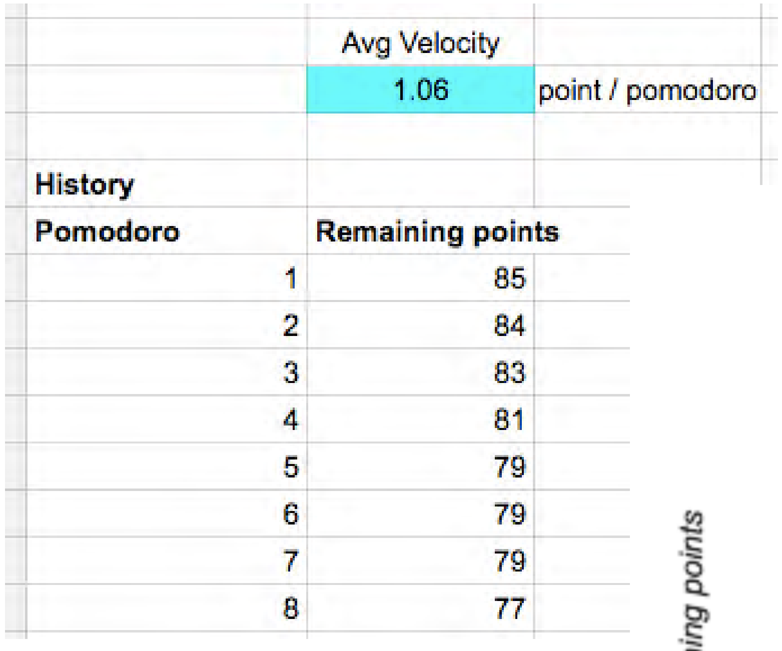


Measure:

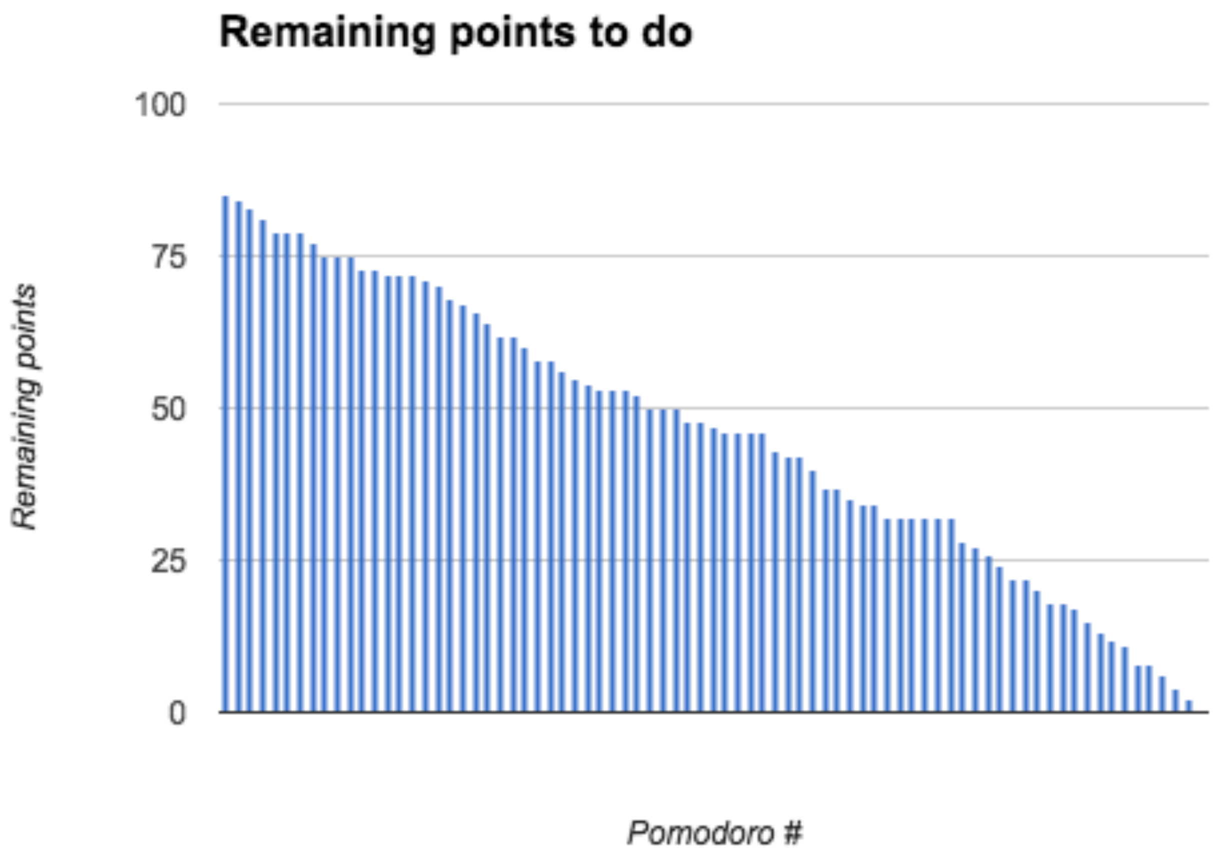
- How much can I get done in one Pomodoro?
- How many Pomodoros can I execute per day / week?

Henrik Kniberg

Used Yesterday's Weather and burndown chart to reliably forecast when the video would be done



	Avg Velocity	
	1.06	point / pomodoro
History		
Pomodoro	**Remaining points**	
1	85	
2	84	
3	83	
4	81	
5	79	
6	79	
7	79	
8	77	



Henrik Kniberg

Strategies for applying agile in other contexts

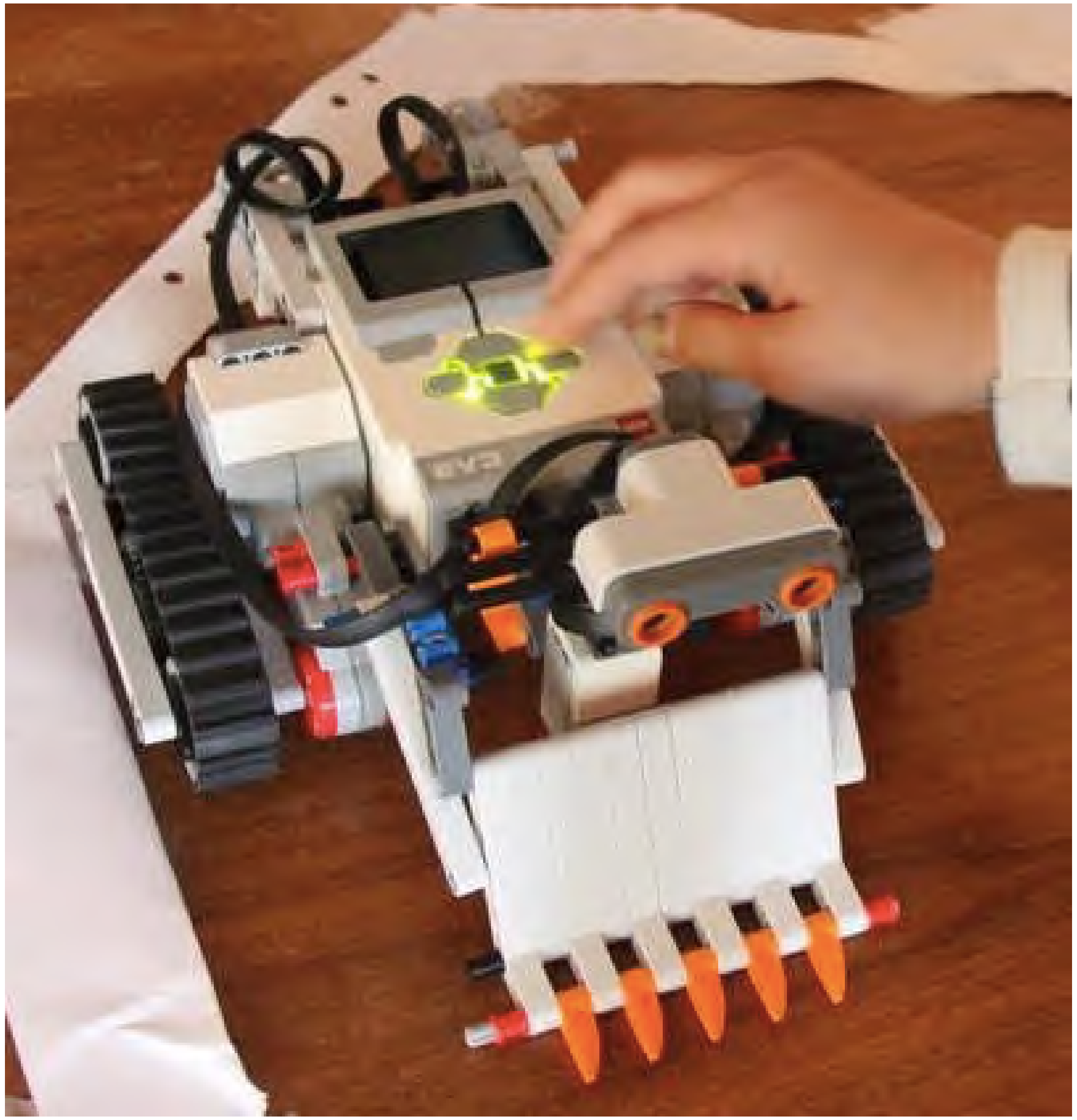
Implement method X "by the book", and follow the rules religously

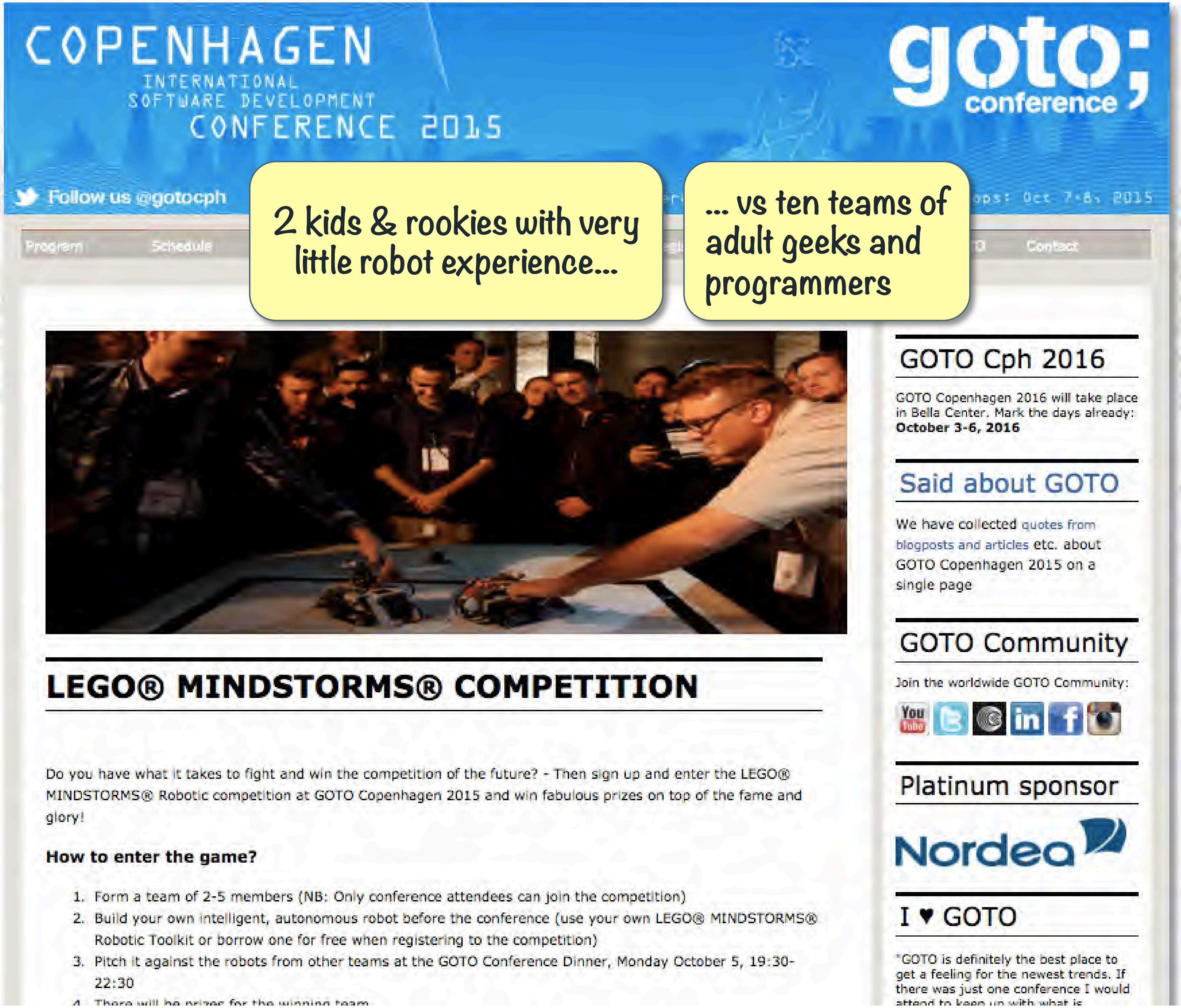
Implement method X "by the book", then customize it

Cherry-pick specific practices

Apply agile ideas directly, without using any specific framework

Robit





2 kids & rookies with very little robot experience...

... vs ten teams of adult geeks and programmers

GOTO Cph 2016

GOTO Copenhagen 2016 will take place in Bella Center. Mark the days already: **October 3-6, 2016**

Said about GOTO

We have collected quotes from blogposts and articles etc. about GOTO Copenhagen 2015 on a single page

GOTO Community

Join the worldwide GOTO Community:

Platinum sponsor

Nordea

I ♥ GOTO

"GOTO is definitely the best place to get a feeling for the newest trends. If there was just one conference I would attend to keep up with what is

LEGO® MINDSTORMS® COMPETITION

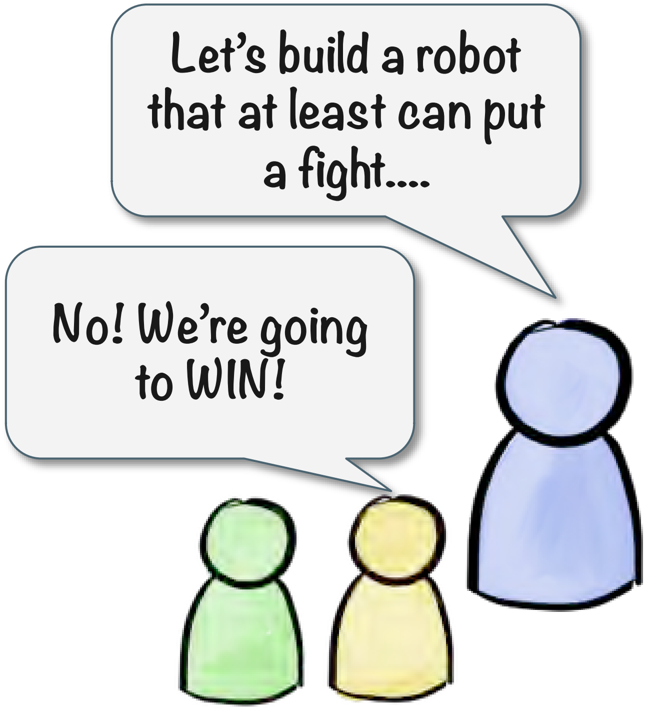
Do you have what it takes to fight and win the competition of the future? - Then sign up and enter the LEGO® MINDSTORMS® Robotic competition at GOTO Copenhagen 2015 and win fabulous prizes on top of the fame and glory!

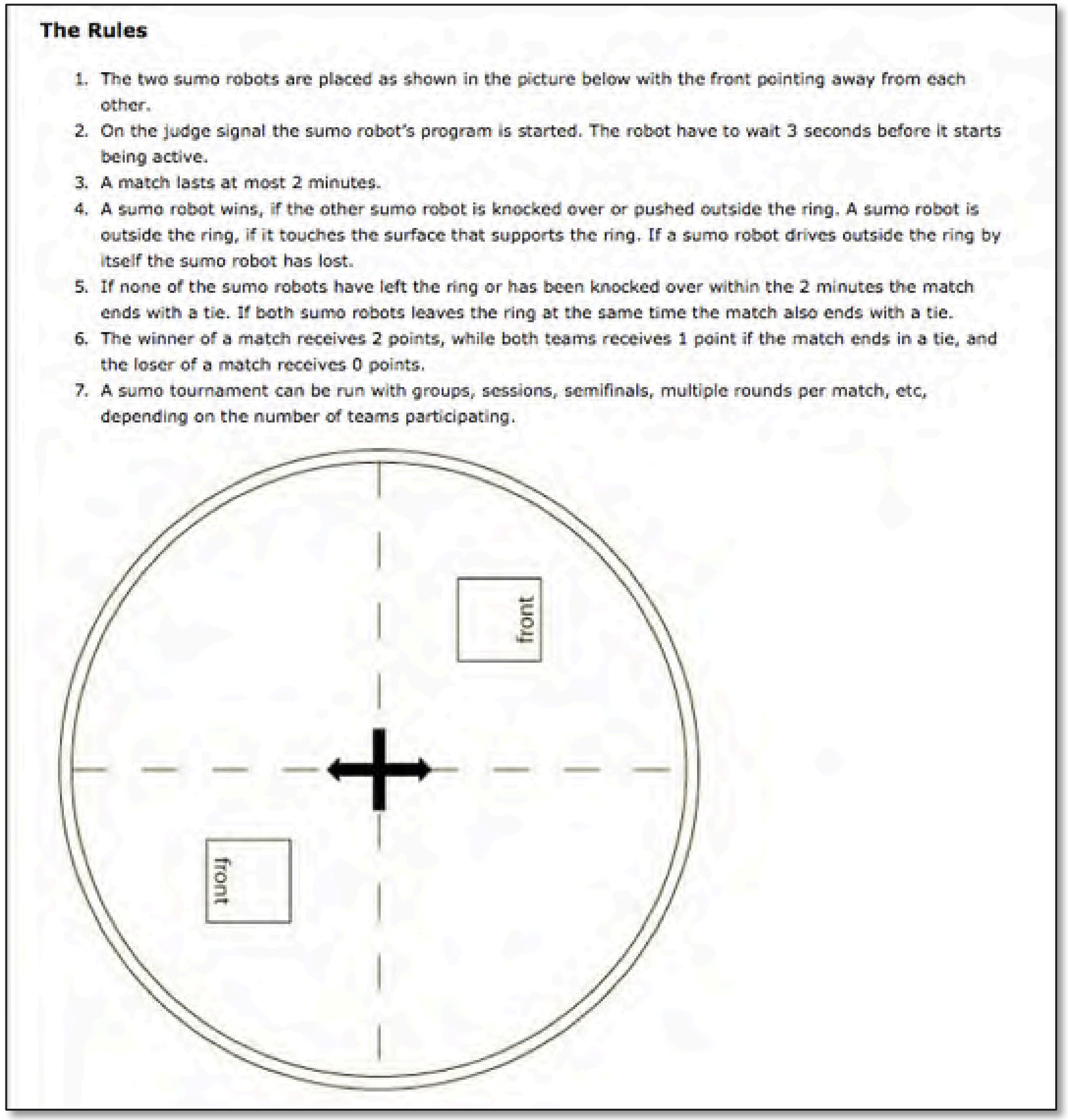
How to enter the game?

1. Form a team of 2-5 members (NB: Only conference attendees can join the competition)
2. Build your own intelligent, autonomous robot before the conference (use your own LEGO® MINDSTORMS® Robotic Toolkit or borrow one for free when registering to the competition)
3. Pitch it against the robots from other teams at the GOTO Conference Dinner, Monday October 5, 19:30-22:30
4. There will be prizes for the winning team

Henri

Step 1: Set a clear goal (define "success")





The Rules

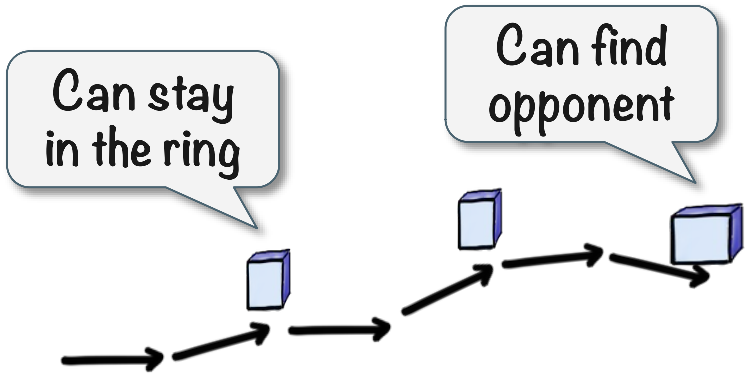
1. The two sumo robots are placed as shown in the picture below with the front pointing away from each other.
2. On the judge signal the sumo robot's program is started. The robot have to wait 3 seconds before it starts being active.
3. A match lasts at most 2 minutes.
4. A sumo robot wins, if the other sumo robot is knocked over or pushed outside the ring. A sumo robot is outside the ring, if it touches the surface that supports the ring. If a sumo robot drives outside the ring by itself the sumo robot has lost.
5. If none of the sumo robots have left the ring or has been knocked over within the 2 minutes the match ends with a tie. If both sumo robots leaves the ring at the same time the match also ends with a tie.
6. The winner of a match receives 2 points, while both teams receives 1 point if the match ends in a tie, and the loser of a match receives 0 points.
7. A sumo tournament can be run with groups, sessions, semifinals, multiple rounds per match, etc, depending on the number of teams participating.

Agile
I'M GOING TO HAVE TO ~~SCIENCE~~
THE SHIT OUT OF THIS

Step 2: Build a Minimum Viable Robot (Earliest Testable Robot)

Henrik Kniberg

Aim for the clouds, but deliver and test in small steps

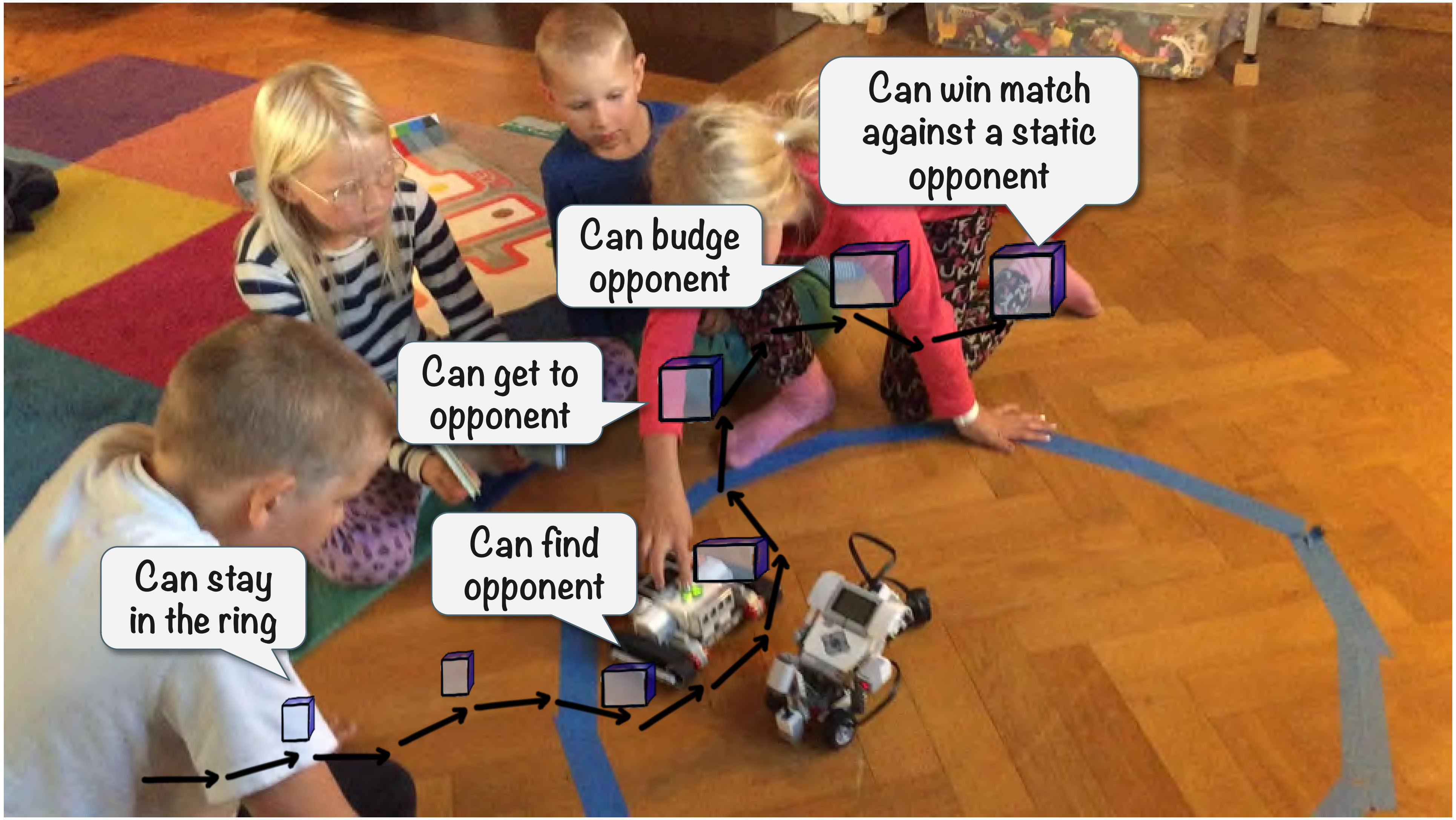


Henrik Kniberg

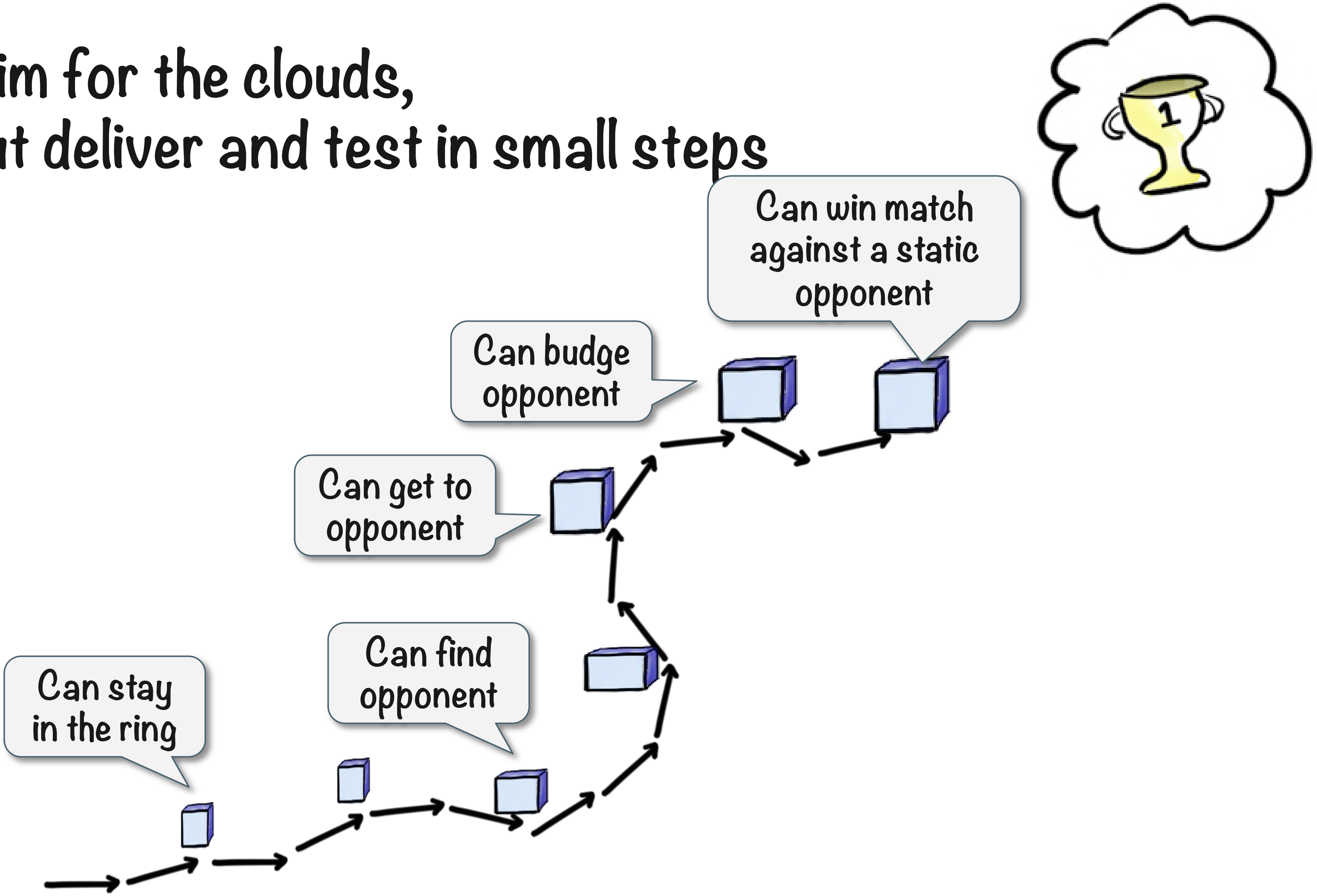
Step 3: Build an opponent to practice against

Henrik Kniberg

Field test, Field test, Field test



Aim for the clouds, but deliver and test in small steps

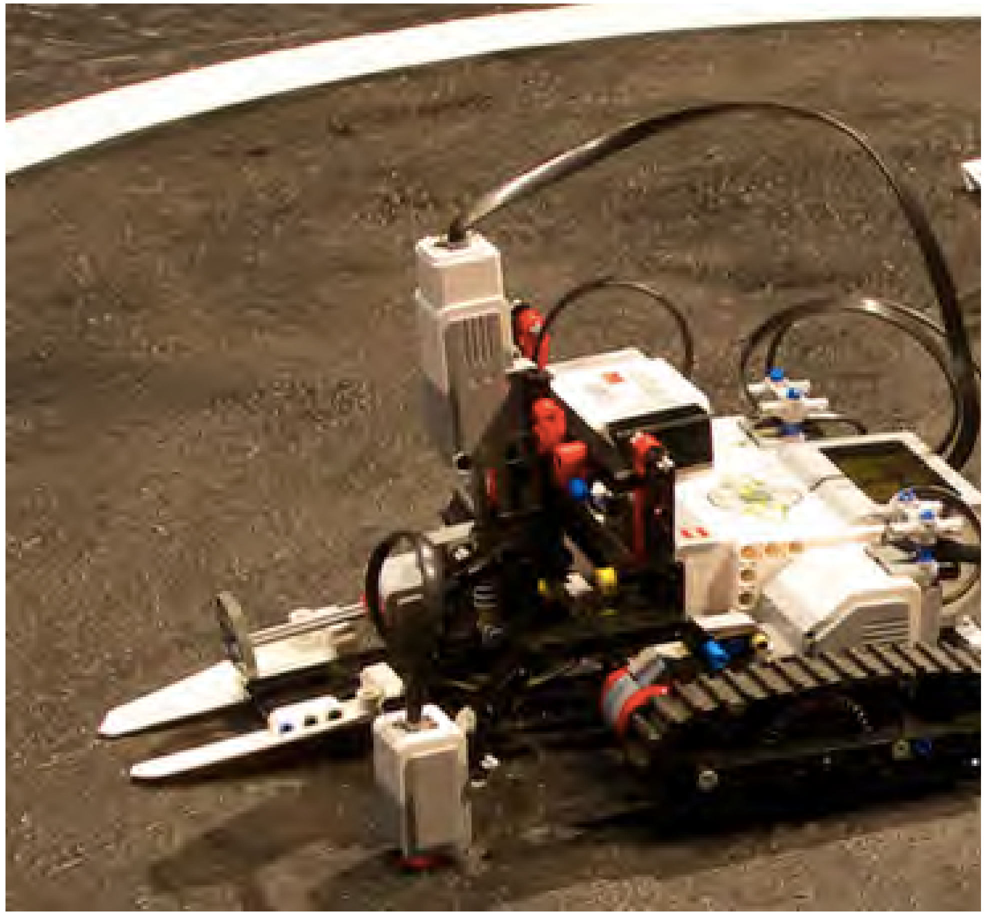


Lifter? Or no lifter?

Hypothesis:
- Mechanical Lifter can help us win

Experiment:
- Build a simple lifter and try

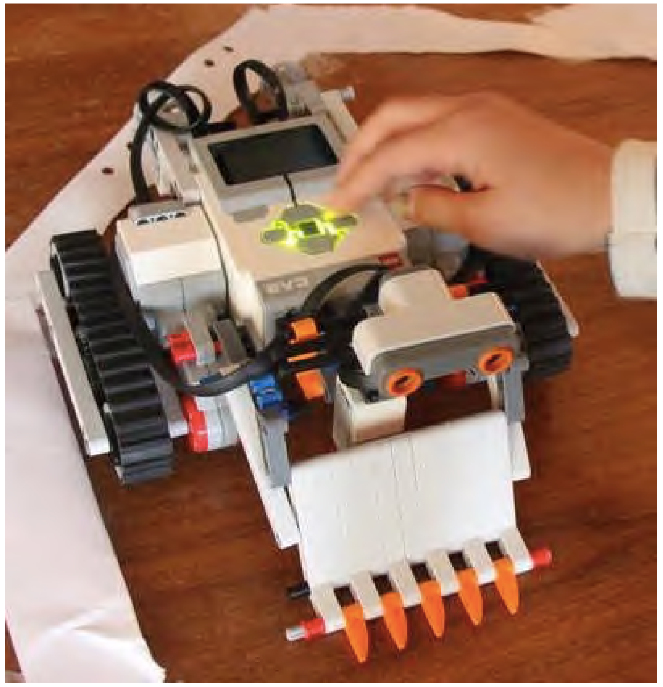
Learning:
- Works as designed...
- But too weak to lift opponent
- ... so it doesn't help us win!



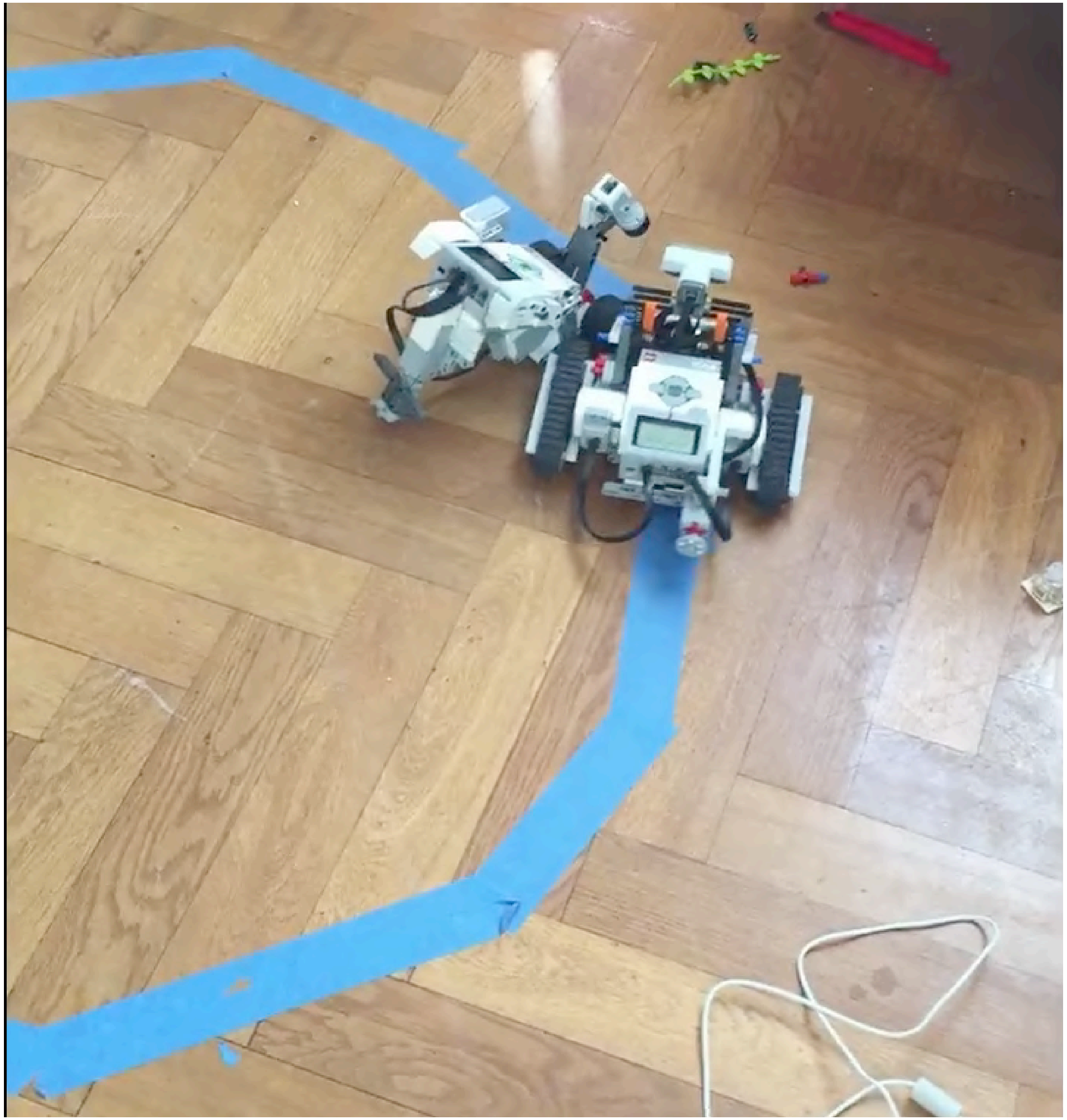
Options:
- Keep it cuz it's cool (who needs to win anyway)
- Improve it
- Remove it, try a different approach

Henrik Kniberg

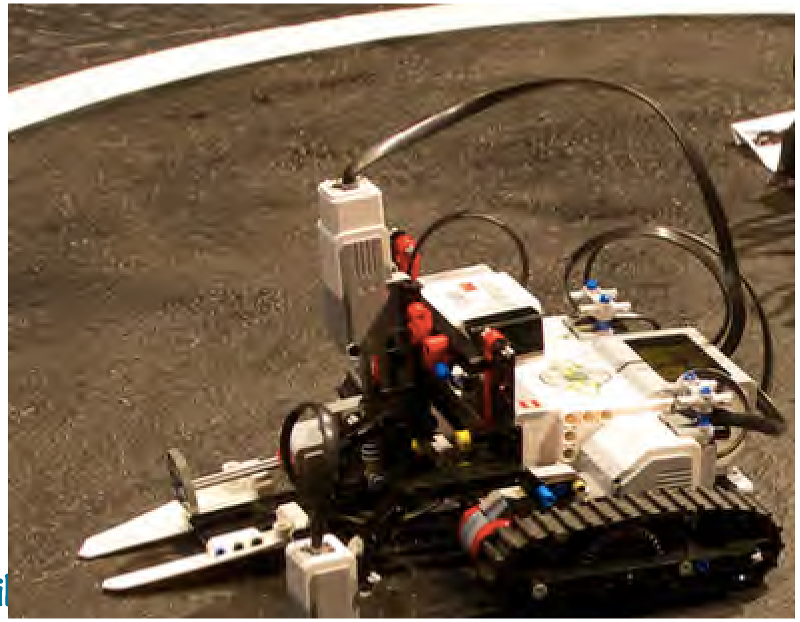
Simpler was better



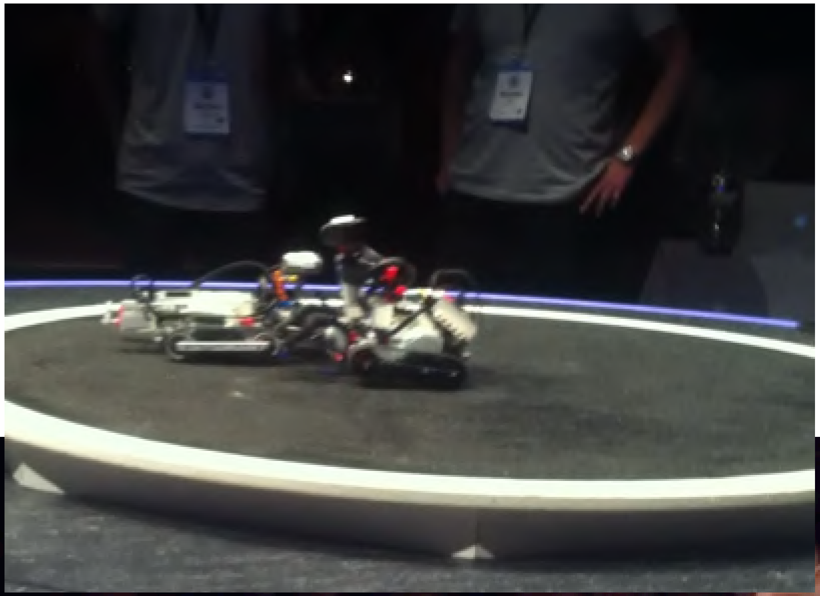
Field testing = Success by 100 failures



Henrik Kniberg



Henri



Henrik Kniberg

Loser of J (if necessary)

How could they win?

Building skill? No.
Programming skills? No.
Luck? Partly, but not entirely.

1) Clear goal
2) Low self-confidence
3) Emergent design
4) LOTS of field testing!

Henrik Kniberg

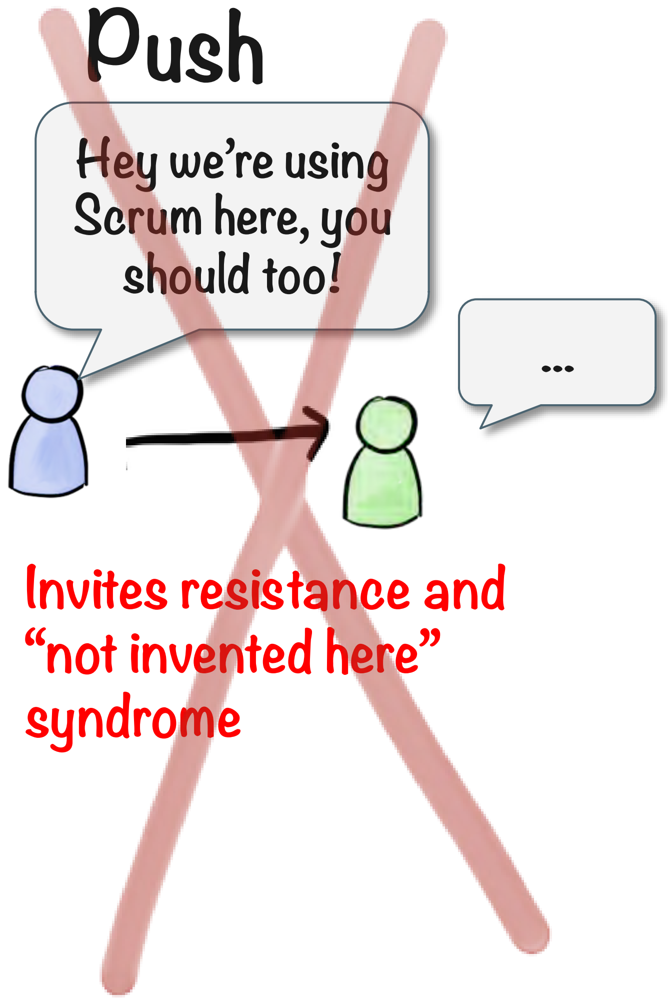
Some tips when applying agile in <insert domain here>

Henrik Kniberg

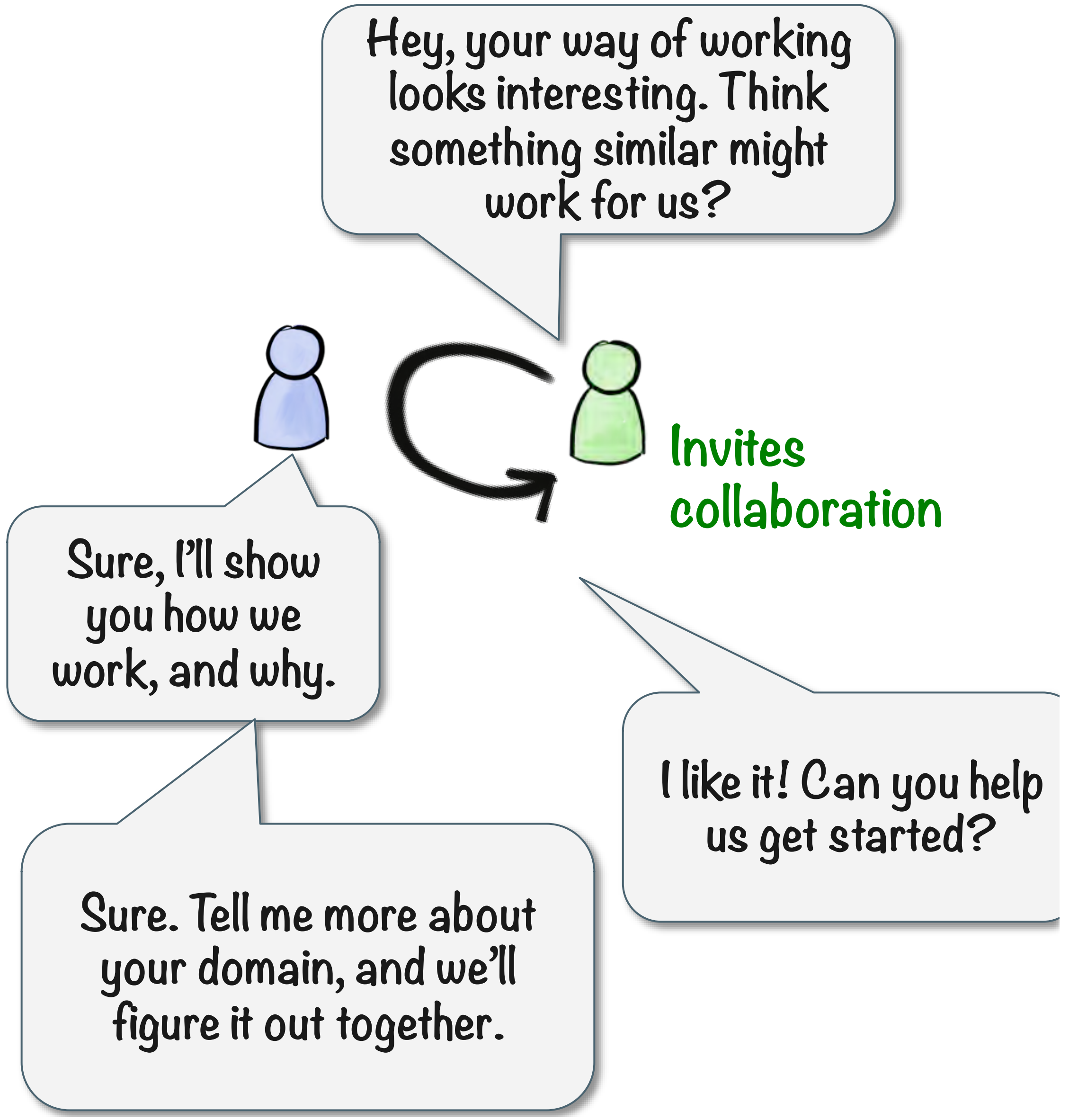
Don't inflict help on people.

Pull works better than Push

Push


Hey we're using Scrum here, you should too!
...
Invites resistance and "not invented here" syndrome

Pull


Hey, your way of working looks interesting. Think something similar might work for us?
Sure, I'll show you how we work, and why.
I like it! Can you help us get started?
Sure. Tell me more about your domain, and we'll figure it out together.
Invites collaboration

Henrik Kniberg

WARNING

2 slides full of bullet points coming up

sorry...

Agile in Domain X requires a collaboration between people who understand Domain X, and people who understand Agile.

Step 1: Understand the context

- What do you do?
- Who are your stakeholders?
- What is a unit of work?
- What does Done mean?
- What does Success look like?
- Who is need to get things to Done?
- What do you want to improve, and why?
- How will you know if you've improved?

Step 2: Understand the tools

- What is Agile? Scrum? Kanban? XYZ?
- Which principles and practices are most applicable in your context?

Step 3: Get Buy-in

- Who needs to be involved to make the change happen?
- What's in it for them?

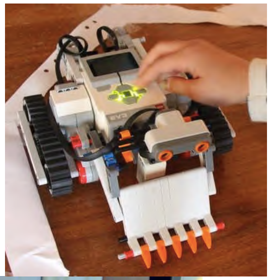
Step 4: Start experimenting

- When in doubt, start by making work visible
- Find some early wins to build trust

Henrik Kniberg

Take-aways

- Agile is not new, and not going away
  - The word may go out of fashion, but the ideas are timeless
- Agile can be useful in just about any context, not just software
  - But Agile or <insert framework here> is only a means, never a goal
- Distinguish between Principles and Practices
  - Practices are more domain-specific and need to be adapted or replaced
- Copy & Paste & Evolve
  - No need to reinvent the wheel
- Use the appropriate language for the domain
  - Don't unnecessarily alienate people with strange words
- Don't inflict help on people
  - If they are happy with their current way of working, then don't bother trying to change it.



Henrik Kniberg