DAT257

Lecture 2: Working in agile teams

Jonas Petrén

Welcome! We will begin in a few minutes

Agenda

- Working in teams
  - Social contract
  - Cross-functional teams
- Agile Planning
  - Task boards
  - Estimation
- How to know how you are doing?
  - Burn-up chart
  - Burn-down chart
  - Velocity
  - Other metrics
- Ending

Have you ever...?

Had a great evening out with friends without pre-booking all the activities?

Planned a family event at a high level, and left the little details to the end?

Found a cheap and easy way to try a new hobby, and then learned it's not for you?

Ran an idea by a friend before fully forming it, and changed the idea as a result of the conversation?

Congratulations!
This is agile!

Jonas Petrén

10 years in software testing, development and as test manager. **6+ years of Scrum Master experience.** Full-time Scrum Master since beginning of 2018. Employed by HiQ since 2012.

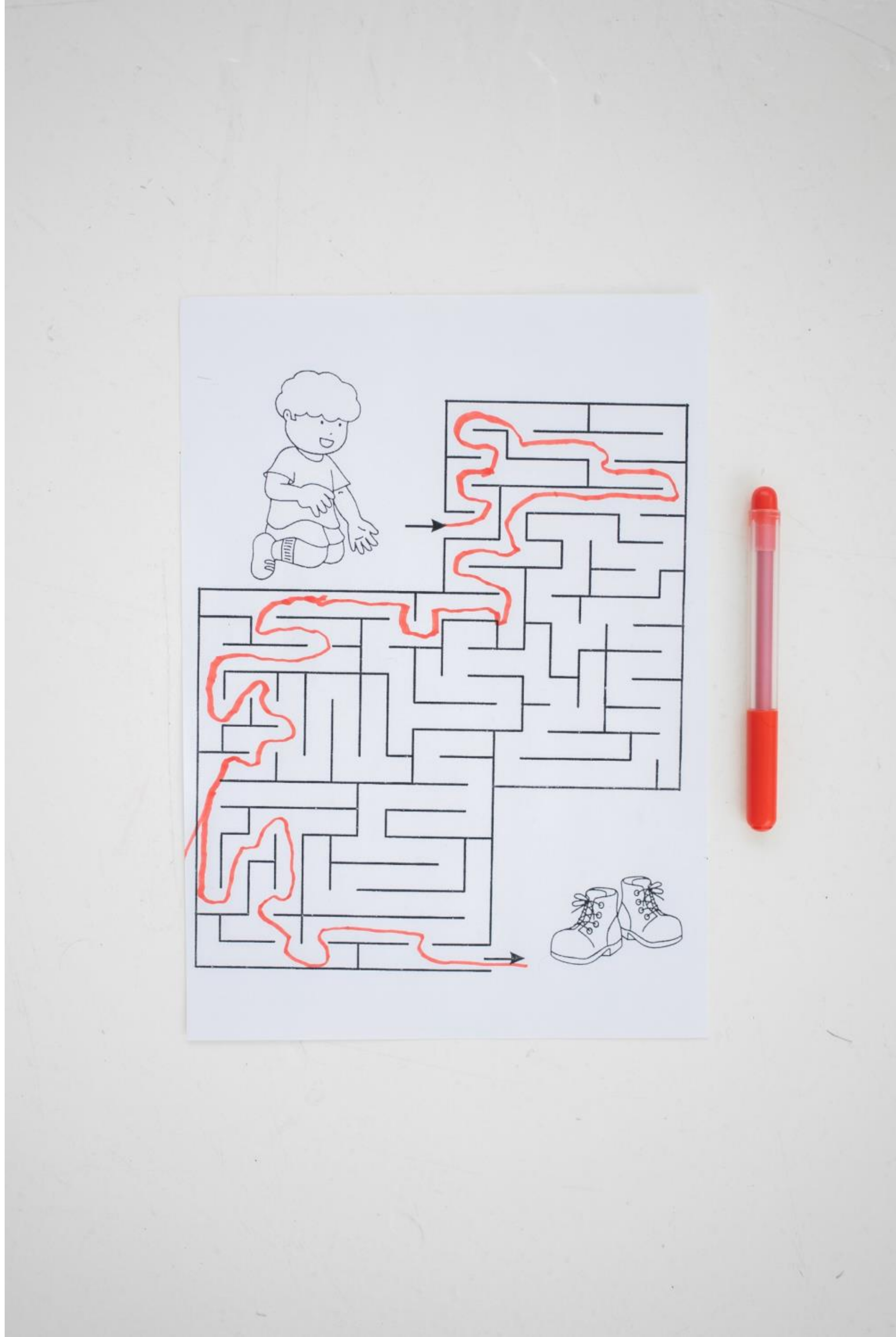
Civilingenjör Informationsteknologi, Linköping University, 2003-2008

Today: Senior Scrum Master/Agile coach

Certified **Professional Scrum Master III**

The goal for today

To get an understanding of how it is to work in an agile team



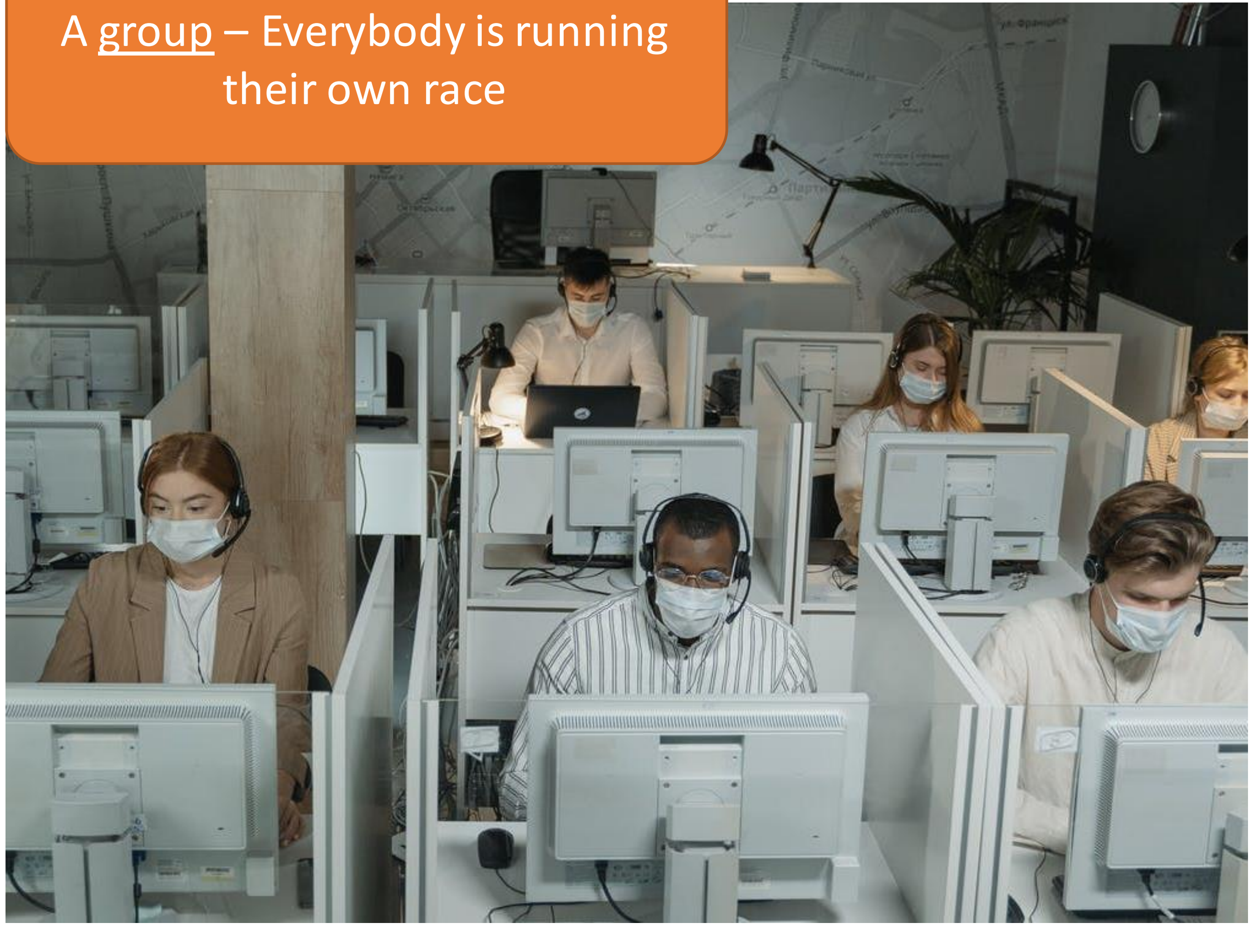
Working in teams

"The way a team plays as a whole determines its success. You may have the greatest bunch of individual stars in the world, but if they don't play together, the club won't be worth a dime." – Babe Ruth

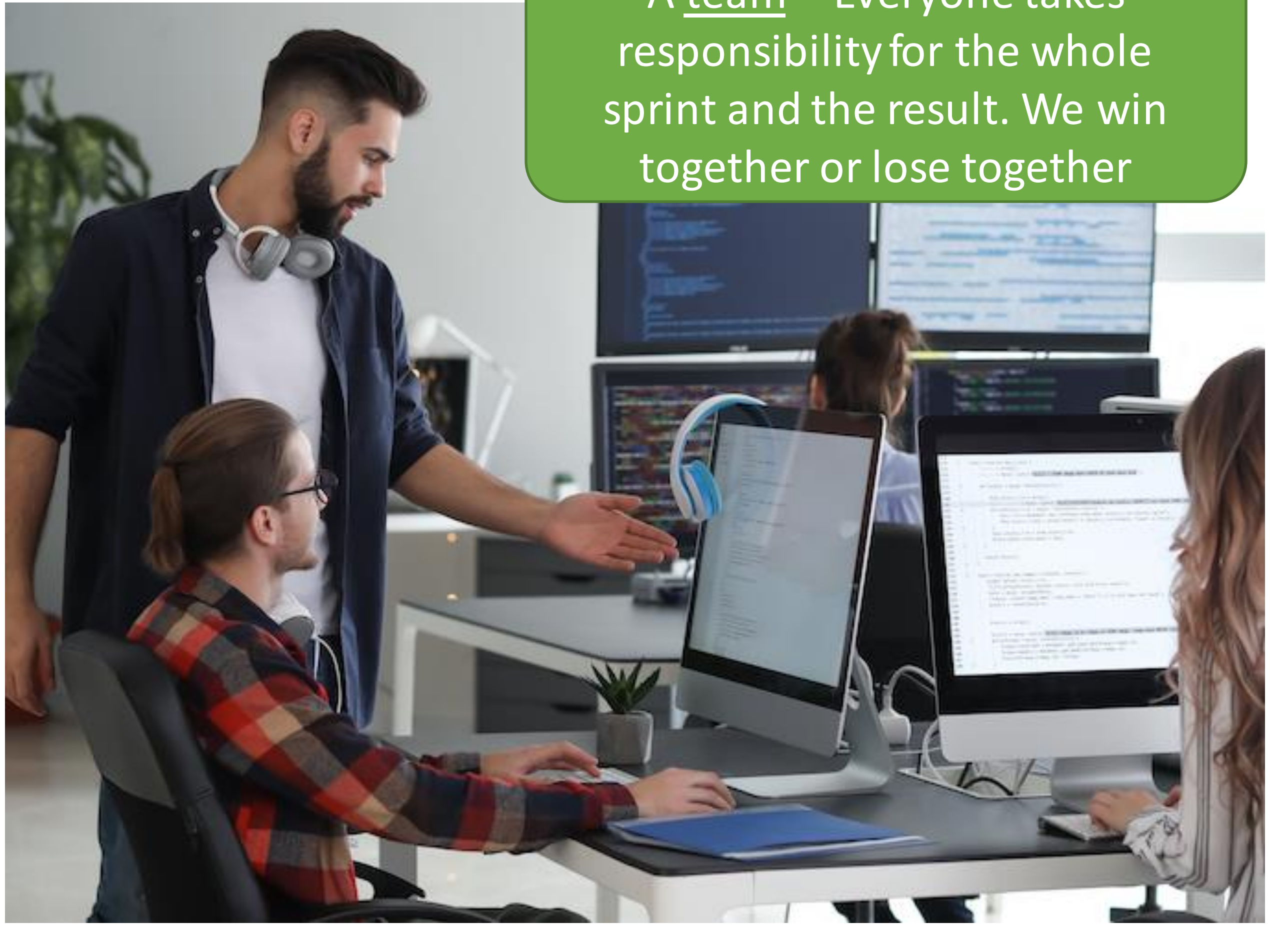
A group – Everybody is running their own race

A team – Everyone takes responsibility for the whole field and the result. We win together or lose together

A group – Everybody is running their own race



A team – Everyone takes responsibility for the whole sprint and the result. We win together or lose together



Why teams?

We work with **COMPLEX PROBLEMS**. Many brains think better than one.

A team **TACKLES CHALLENGES** instead of just coping with them

Without teams we have **HANDOVERS** which results in information loss

On the personal front: teams result in faster **PROFESSIONAL GROWTH** and greater **ENGAGEMENT**

Social contract or working agreement

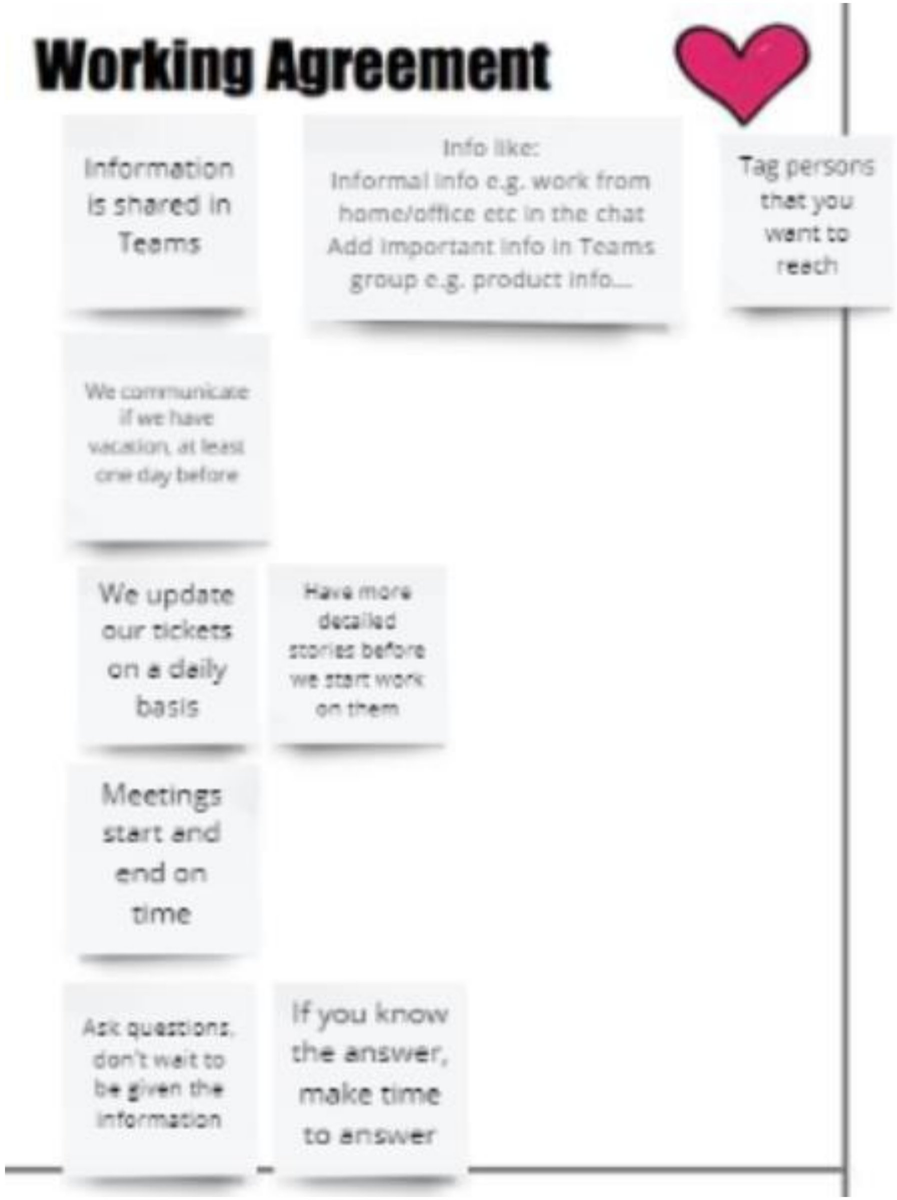
Contains your team's agreement on how to work together

Make the rules explicit so that everyone knows how to act. No unspoken rules

Enfores good behaviour

Hold each other to account for keeping the agreements

Make sure everyone in the team can stand behind the rules and commit to them. If not, you set yourself up for failure immediately



Working Agreement

Information is shared in Teams

Info like:
Informal info e.g. work from home/office etc in the chat
Add important info in Teams group e.g. product info...

Tag persons that you want to reach

We communicate if we have vacation, at least one day before

We update our tickets on a daily basis

Have more detailed stories before we start work on them

Meetings start and end on time

Ask questions, don't wait to be given the information

If you know the answer, make time to answer

Example from Jonas' previous team

- Information is shared in Teams
- We communicate if we have vacation, at least one day before
- Meetings start and end on time
- Ask questions, don't wait to be given information
- If you know the answer, make time to answer

Story: Fikapinnar

If someone broke the agreements e.g. came late to a meeting he/she got a "fikapinne" and when someone reached ten "fikapinnar" he/she would buy fika to the team

Takeaway: This may work very well in a certain team but may not work in another

Other examples of social contract agreements

We are available between 9:00 – 15:00, unless the team is notified

Everybody's opinion should be heard

There are no stupid ideas

No electronic device distractions

Make decisions by consensus

No team commuication after 6pm or on weekends

We always do pair-programming

What you should do this week

Create your team's social contract. Maximum five things

Write them in a text file and put it in your team repository

Consider these areas:

- How do we keep each other informed?
- How do we behave when we meet (physically and digitally)?
- How do we make decisions?

Don't rush! Take time to discuss so that everyone can commit to all rules

Discussion 2 minutes

Turn to your seat neighbour. Discuss the first thing that you would like to add to your team's social contract

Cross-functional teams

"Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint."

This is not the same as *"everyone should be able to do everything"* or *"we don't need specialists"*

It just means that the team should have all skills needed to take a task and turn it in to software/product. They are not dependent on people outside the team.

And we definitely need specialists

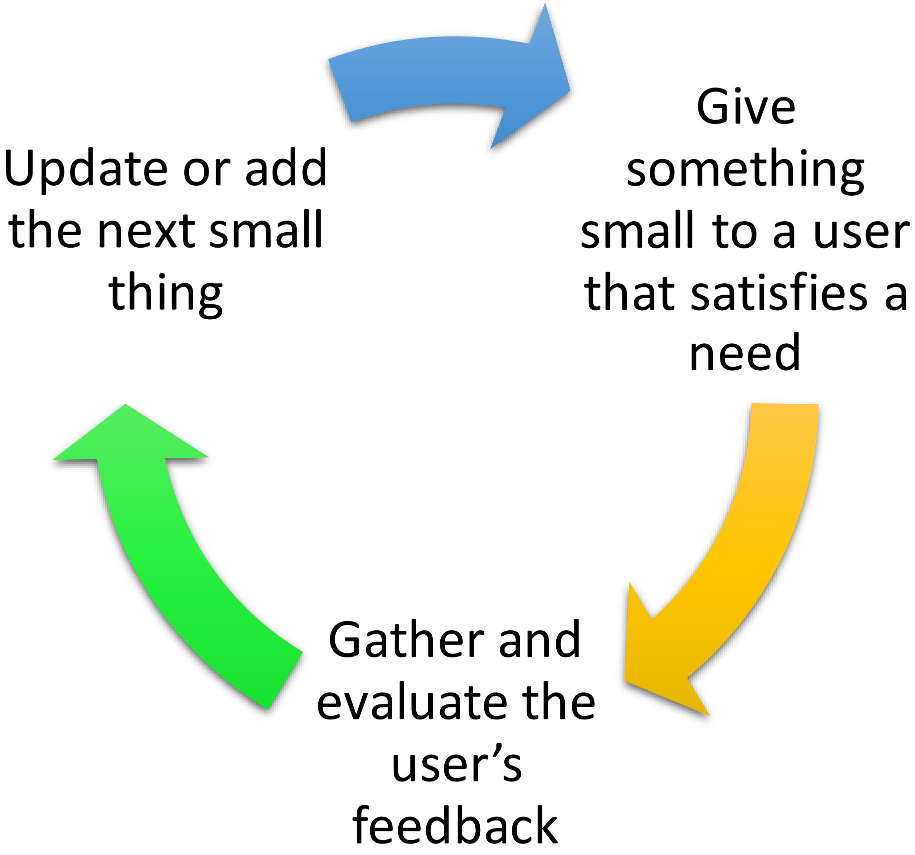
Agile planning

“Agile planning is when we decide together, every little while, what are the best things to do next so that we achieve what we want at the end.” Gil Broza

Setting the expectations right when working in agile

The expectation is not to deliver a big chunk of work six months from now

The expectation is to deliver **SMALL THINGS OFTEN** and to **CONTINUOUSLY LEARN** what the users need and what we will do next to meet their needs

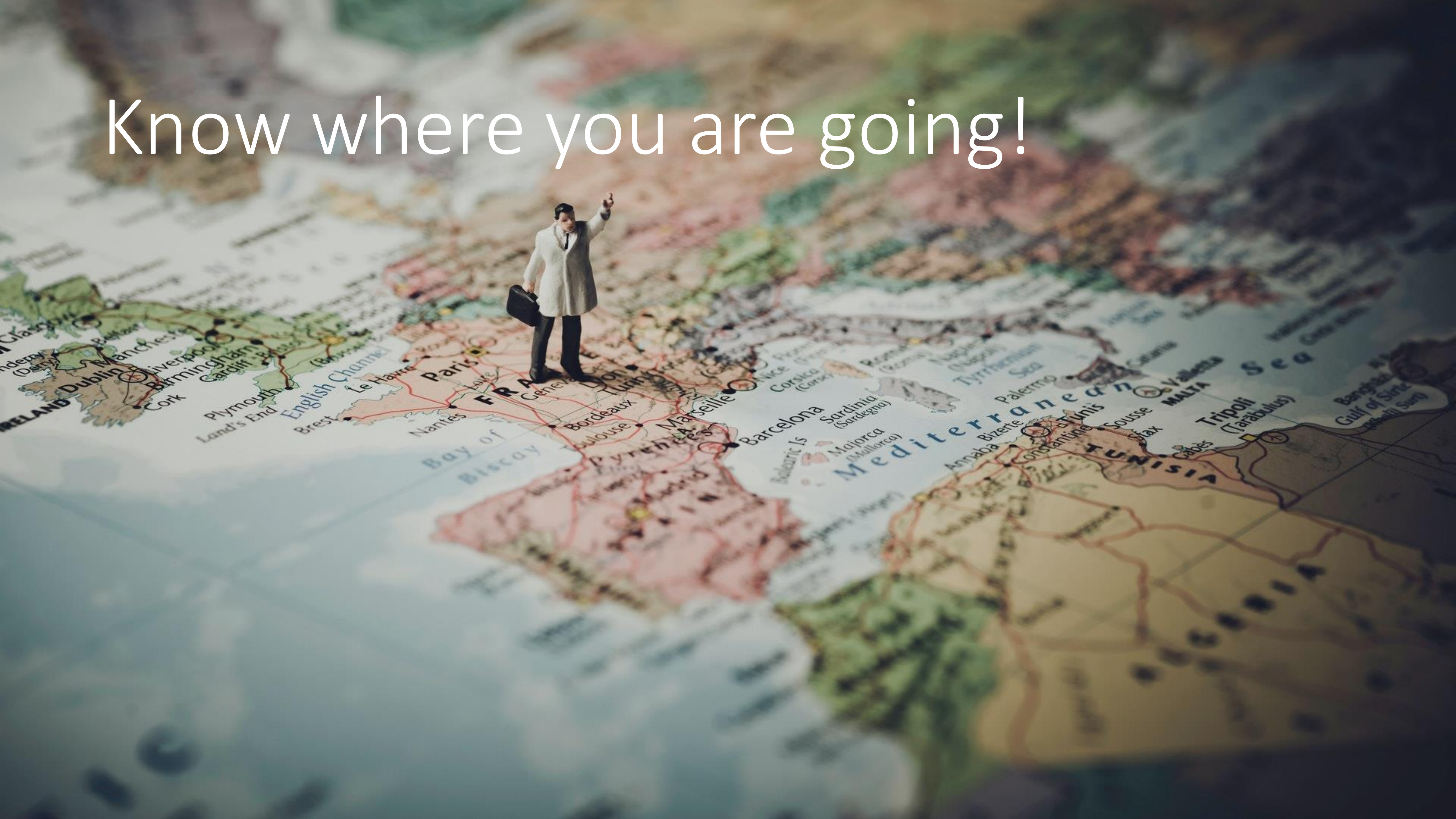


Give something small to a user that satisfies a need

Gather and evaluate the user's feedback

Update or add the next small thing

Know where you are going!



Yesterday's weather is a good prediction of today's weather

The sprint planning:

We do a “forecast”, i.e. a best guess

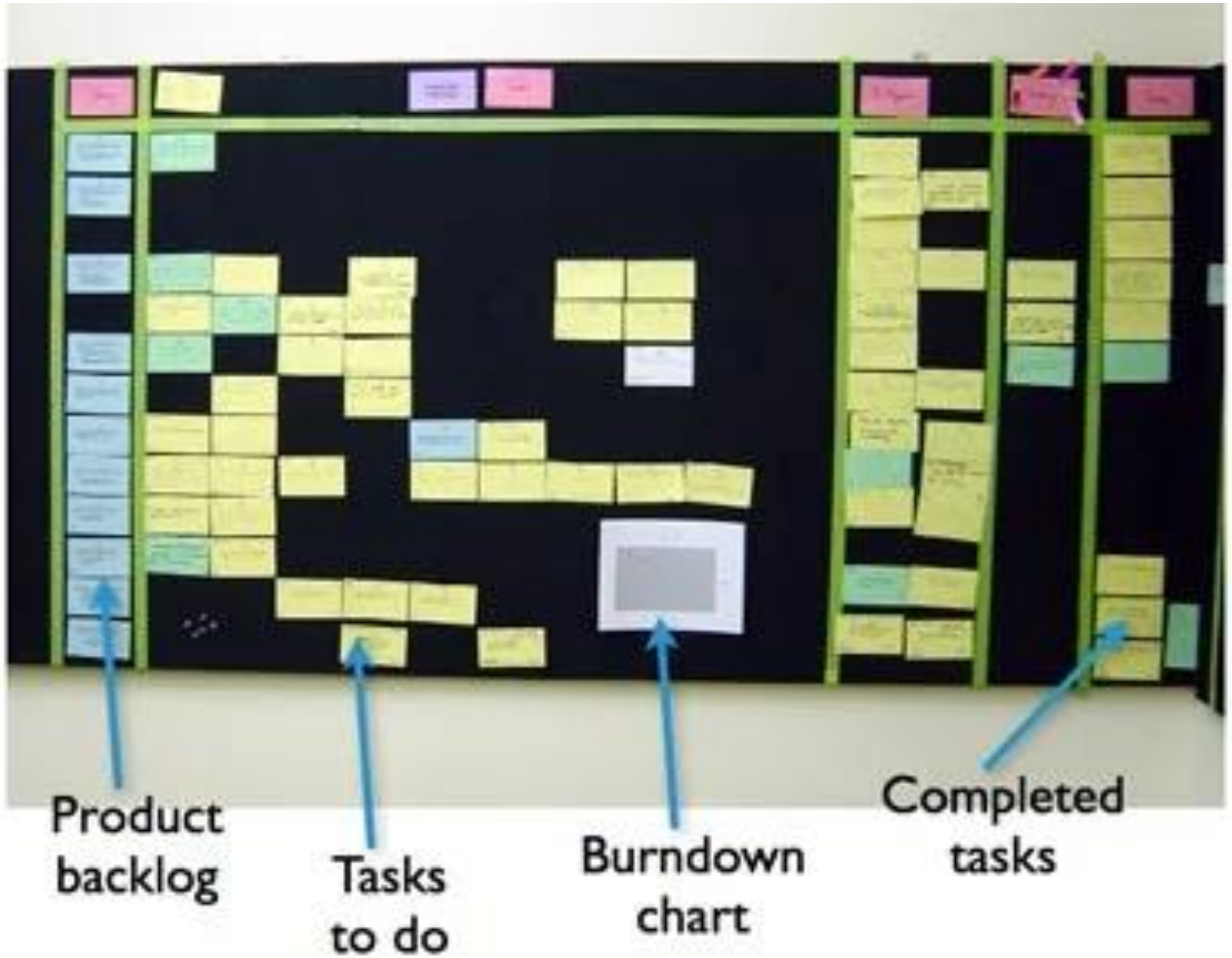
This is not a “target” or ”commitment”

!

Task boards

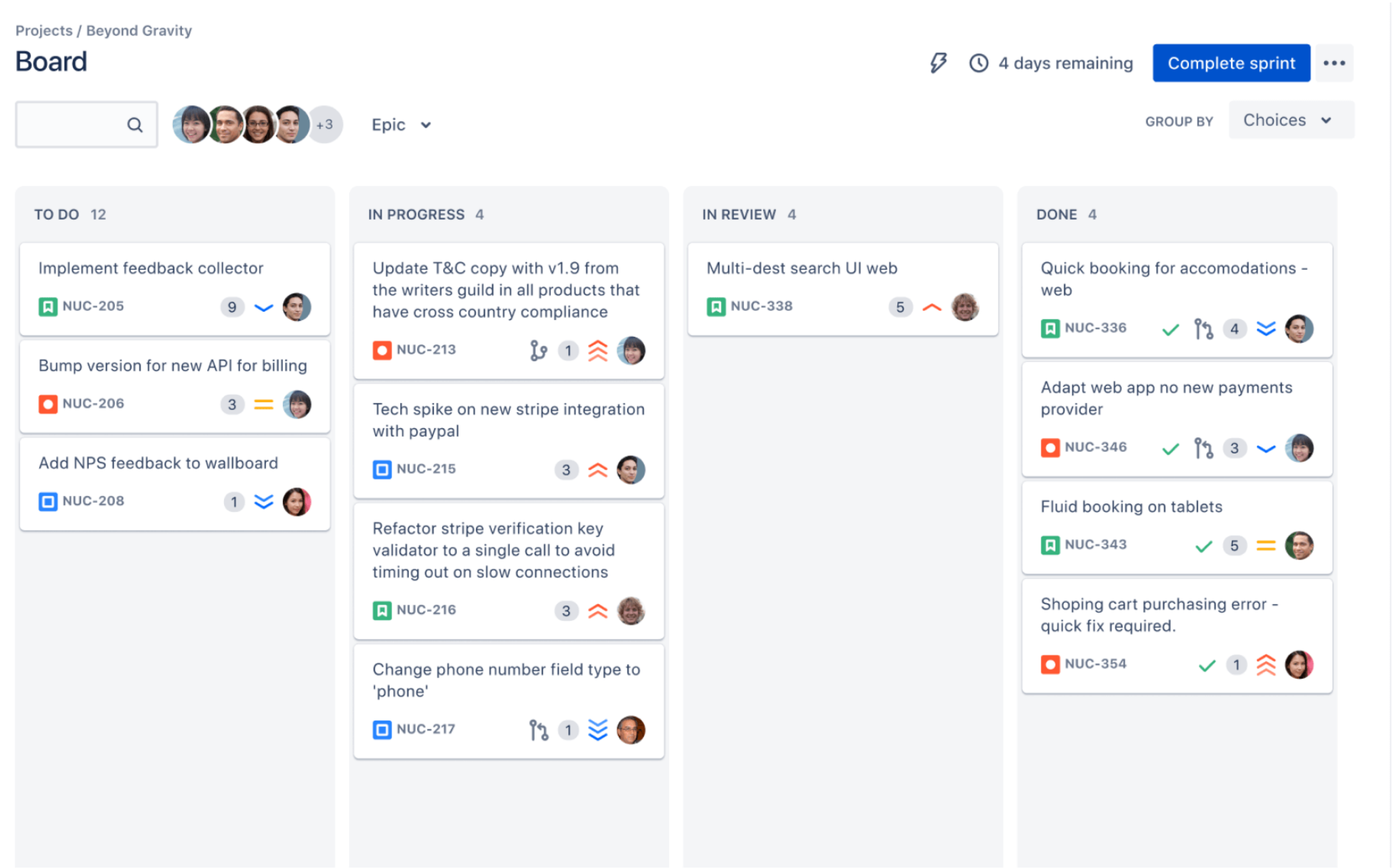
To visualize and keep track of work to be done

How it started – a physical board



Scrum Task Board (mountaingoatsoftware.com)

Jira – the most popular tool



Projects / Beyond Gravity

Board

4 days remaining

Complete sprint

Epic

GROUP BY Choices

TO DO 12

Implement feedback collector
NUC-205 9

Bump version for new API for billing
NUC-206 3

Add NPS feedback to wallboard
NUC-208 1

IN PROGRESS 4

Update T&C copy with v1.9 from the writers guild in all products that have cross country compliance
NUC-213 1

Tech spike on new stripe integration with paypal
NUC-215 3

Refactor stripe verification key validator to a single call to avoid timing out on slow connections
NUC-216 3

Change phone number field type to 'phone'
NUC-217 1

IN REVIEW 4

Multi-dest search UI web
NUC-338 5

DONE 4

Quick booking for accomodations - web
NUC-336 4

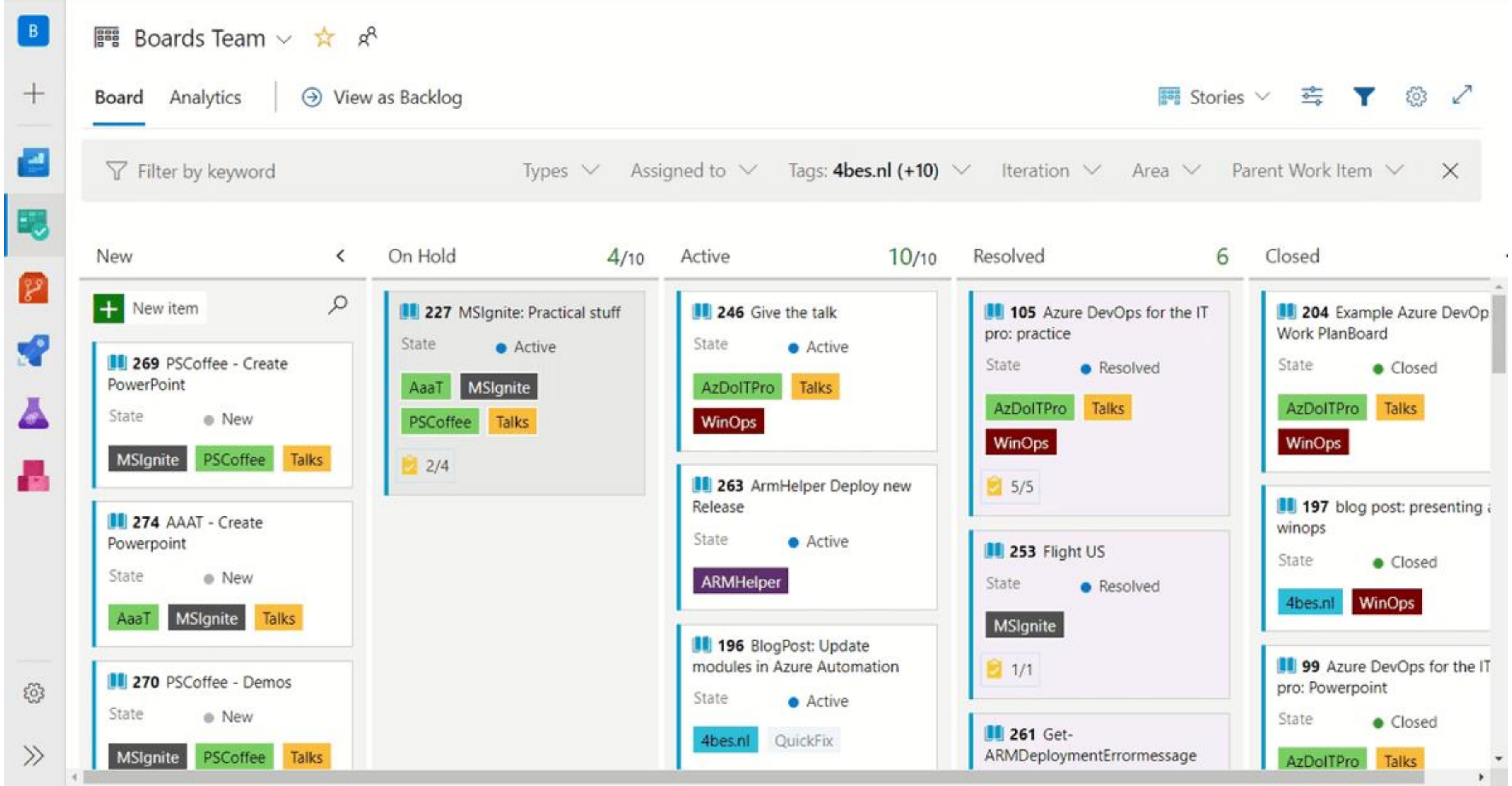
Adapt web app no new payments provider
NUC-346 3

Fluid booking on tablets
NUC-343 5

Shoping cart purchasing error - quick fix required.
NUC-354 1

Jira Scrum Boards | Atlassian

Azure DevOps – also very popular



New

269 PSCoffee - Create PowerPoint
State ● New
MSIgnite PSCoffee Talks

274 AAAT - Create Powerpoint
State ● New
AaaT MSIgnite Talks

270 PSCoffee - Demos
State ● New
MSIgnite PSCoffee Talks

On Hold 4/10

227 MSIgnite: Practical stuff
State ● Active
AaaT MSIgnite PSCoffee Talks
2/4

Active 10/10

246 Give the talk
State ● Active
AzDoITPro Talks WinOps

263 ArmHelper Deploy new Release
State ● Active
ARMHelper

196 BlogPost: Update modules in Azure Automation
State ● Active
4bes.nl QuickFix

Resolved 6

105 Azure DevOps for the IT pro: practice
State ● Resolved
AzDoITPro Talks WinOps
5/5

253 Flight US
State ● Resolved
MSIgnite
1/1

261 Get-ARMDeploymentErrormessage

Closed

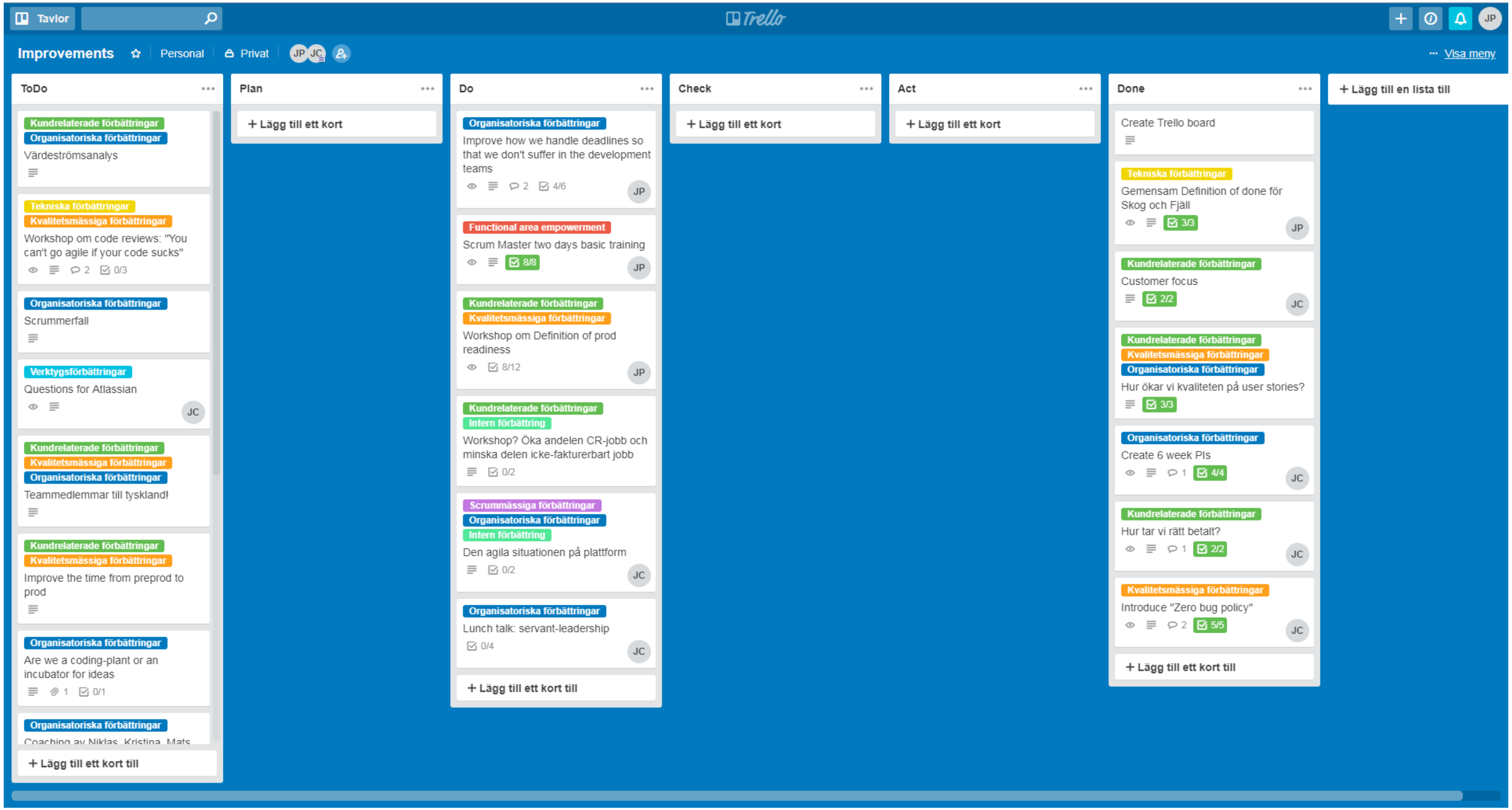
204 Example Azure DevOp Work PlanBoard
State ● Closed
AzDoITPro Talks WinOps

197 blog post: presenting winops
State ● Closed
4bes.nl WinOps

99 Azure DevOps for the IT pro: Powerpoint
State ● Closed
AzDoITPro Talks

[Azure DevOps for the IT Pro: tips & tricks - 4bes.nl](https://4bes.nl)

Trello – easy to use



ToDo

- Kundrelaterade förbättringar
Organisatoriska förbättringar
  Värdeströmsanalys
- Tekniska förbättringar
Kvalitetsmässiga förbättringar
  Workshop om code reviews: "You can't go agile if your code sucks"
  2 0/3
- Organisatoriska förbättringar
  Scrummerfall
- Verktygsförbättringar
  Questions for Atlassian
  JC
- Kundrelaterade förbättringar
Kvalitetsmässiga förbättringar
Organisatoriska förbättringar
  Teammedlemmar till tyskland!
- Kundrelaterade förbättringar
Kvalitetsmässiga förbättringar
  Improve the time from preprod to prod
- Organisatoriska förbättringar
  Are we a coding-plant or an incubator for ideas
  1 0/1
- Organisatoriska förbättringar
  Coaching av Niklas, Kristina, Mats

Plan

Do

- Organisatoriska förbättringar
  Improve how we handle deadlines so that we don't suffer in the development teams
  2 4/6 JP
- Functional area empowerment
  Scrum Master two days basic training
  8/8 JP
- Kundrelaterade förbättringar
Kvalitetsmässiga förbättringar
  Workshop om Definition of prod readiness
  8/12 JP
- Kundrelaterade förbättringar
Intern förbättring
  Workshop? Öka andelen CR-jobb och minska delen icke-fakturerbart jobb
  0/2
- Scrummässiga förbättringar
Organisatoriska förbättringar
Intern förbättring
  Den agila situationen på plattform
  0/2 JC
- Organisatoriska förbättringar
  Lunch talk: servant-leadership
  0/4 JC

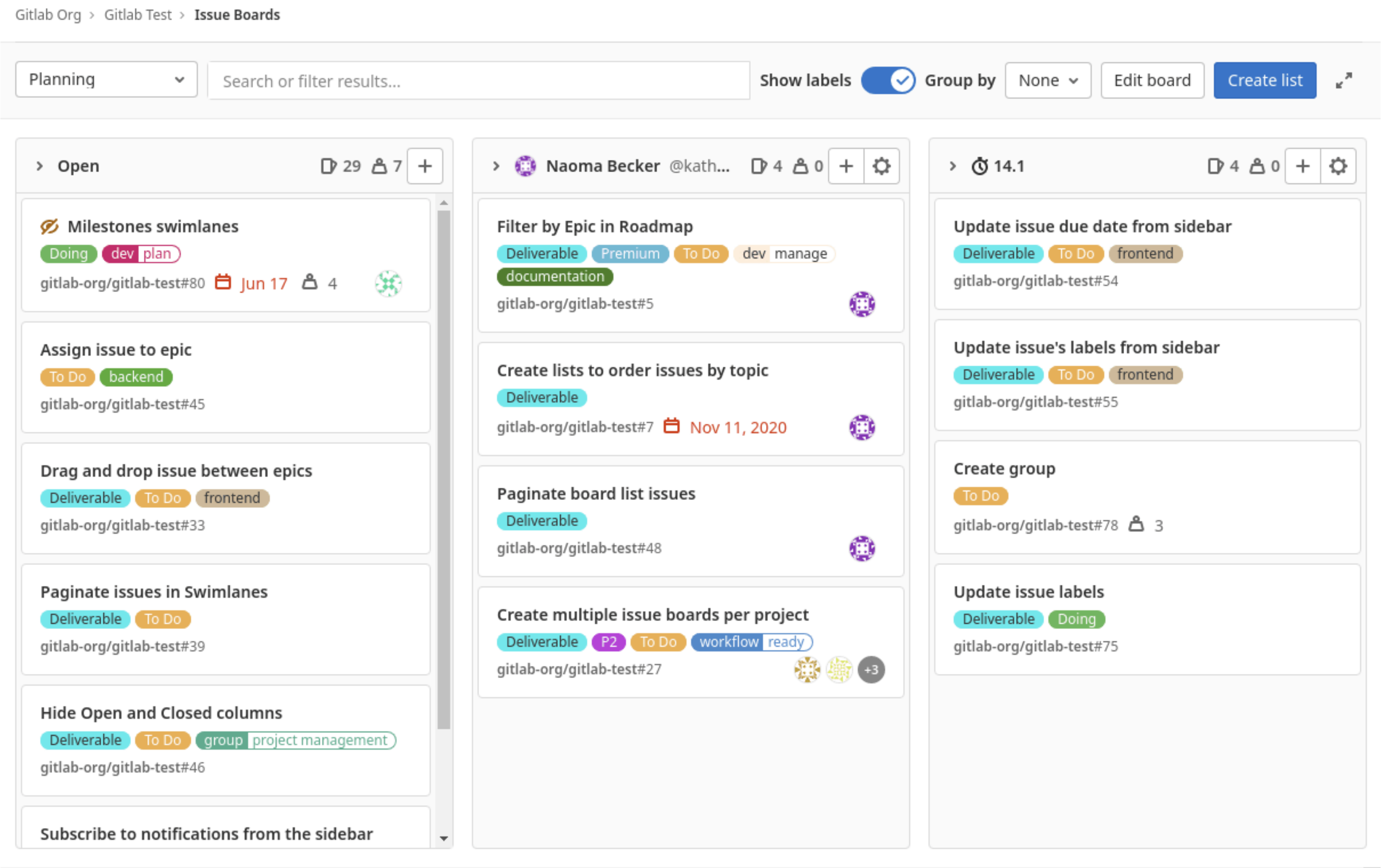
Check

Act

Done

- Create Trello board
- Tekniska förbättringar
  Gemensam Definition of done för Skog och Fjäll
  3/3 JP
- Kundrelaterade förbättringar
  Customer focus
  2/2 JC
- Kundrelaterade förbättringar
Kvalitetsmässiga förbättringar
Organisatoriska förbättringar
  Hur ökar vi kvaliteten på user stories?
  3/3
- Organisatoriska förbättringar
  Create 6 week PIs
  1 4/4 JC
- Kundrelaterade förbättringar
  Hur tar vi rätt betalt?
  1 2/2 JC
- Kvalitetsmässiga förbättringar
  Introduce "Zero bug policy"
  2 5/5 JC

GitLab



Gitlab Org › Gitlab Test › Issue Boards
Planning
Search or filter results...
Show labels
Group by
None
Edit board
Create list

Open 29 7

- Milestones swimlanes — Doing, dev plan — gitlab-org/gitlab-test#80 Jun 17 4
- Assign issue to epic — To Do, backend — gitlab-org/gitlab-test#45
- Drag and drop issue between epics — Deliverable, To Do, frontend — gitlab-org/gitlab-test#33
- Paginate issues in Swimlanes — Deliverable, To Do — gitlab-org/gitlab-test#39
- Hide Open and Closed columns — Deliverable, To Do, group project management — gitlab-org/gitlab-test#46
- Subscribe to notifications from the sidebar

Naoma Becker @kath... 4 0

- Filter by Epic in Roadmap — Deliverable, Premium, To Do, dev manage, documentation — gitlab-org/gitlab-test#5
- Create lists to order issues by topic — Deliverable — gitlab-org/gitlab-test#7 Nov 11, 2020
- Paginate board list issues — Deliverable — gitlab-org/gitlab-test#48
- Create multiple issue boards per project — Deliverable, P2, To Do, workflow ready — gitlab-org/gitlab-test#27 +3

14.1 4 0

- Update issue due date from sidebar — Deliverable, To Do, frontend — gitlab-org/gitlab-test#54
- Update issue's labels from sidebar — Deliverable, To Do, frontend — gitlab-org/gitlab-test#55
- Create group — To Do — gitlab-org/gitlab-test#78 3
- Update issue labels — Deliverable, Doing — gitlab-org/gitlab-test#75

[Issue boards | GitLab]

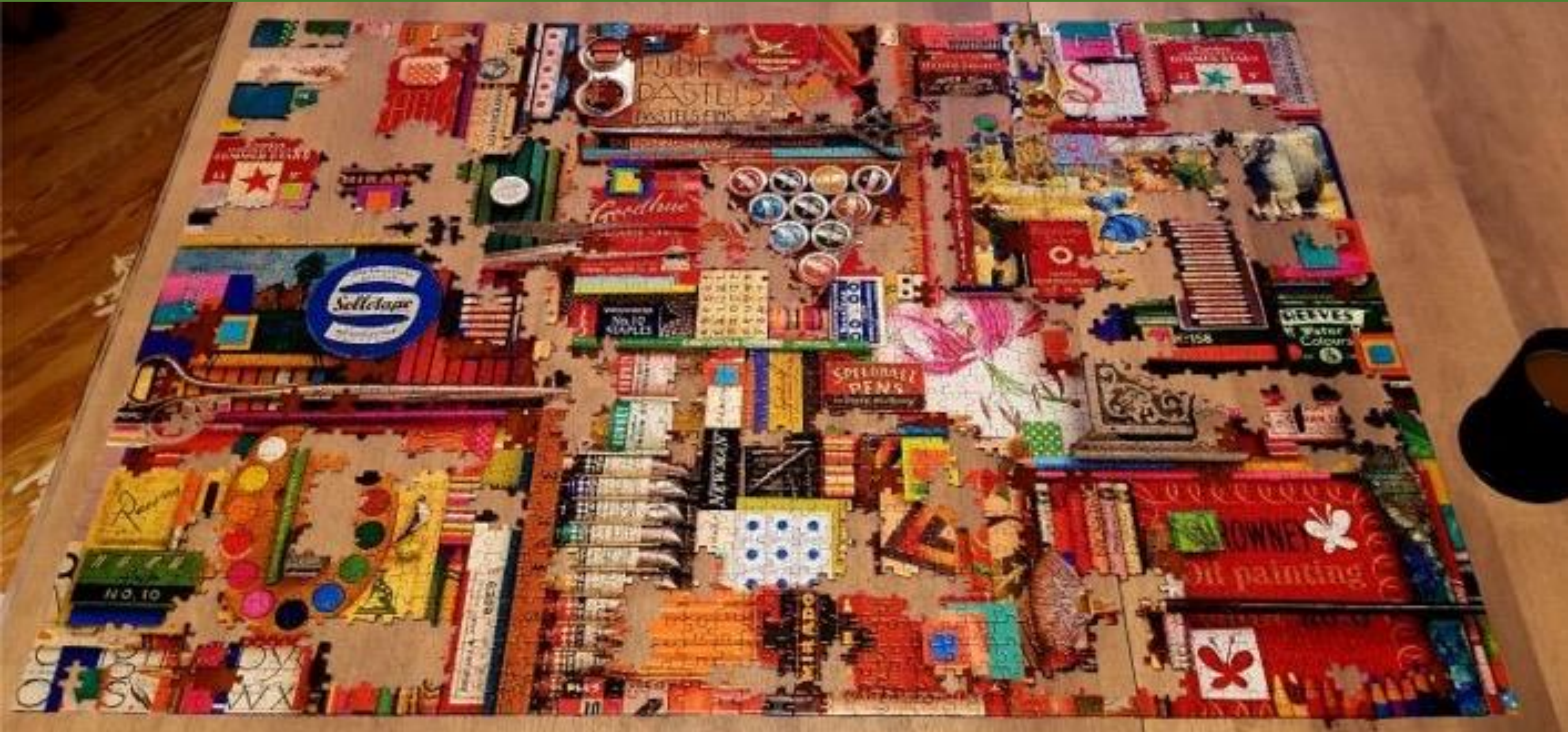
Comparison

Type of board	Advantages	Disadvantages
Physical	Very easy to use. Requires no learning curve	It can only be updated physically, not remotely
Jira	Powerful	Contains much more than you need for this course
Azure DevOps	Powerful	Contains much more than you need for this course. Costs money
Trello	Easy to use	
GitLab	Easily accessed since you are going to work in GitLab with your code	?

Agile estimation

A puzzle with 2000 pieces. How many pieces remain?

Don’t say your answer out loud. Send it to petrenjo@chalmers.se, subject: "Puzzle"



Learnings

Estimating as a group is powerful

Provide ranges, not single numbers

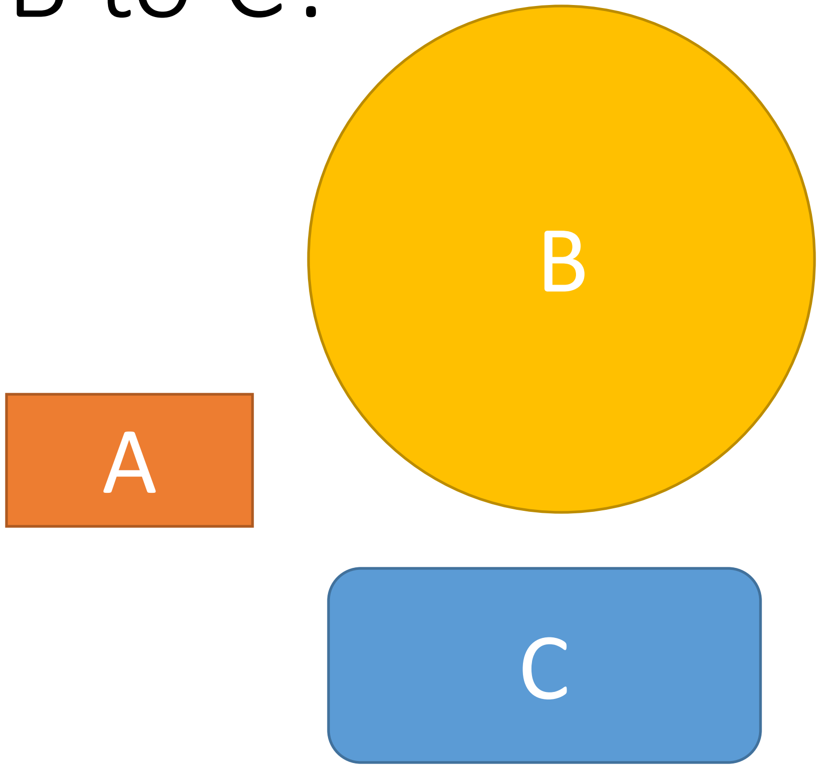
You will (probably) get better over time

✓ **<u>Relative estimation:</u>**

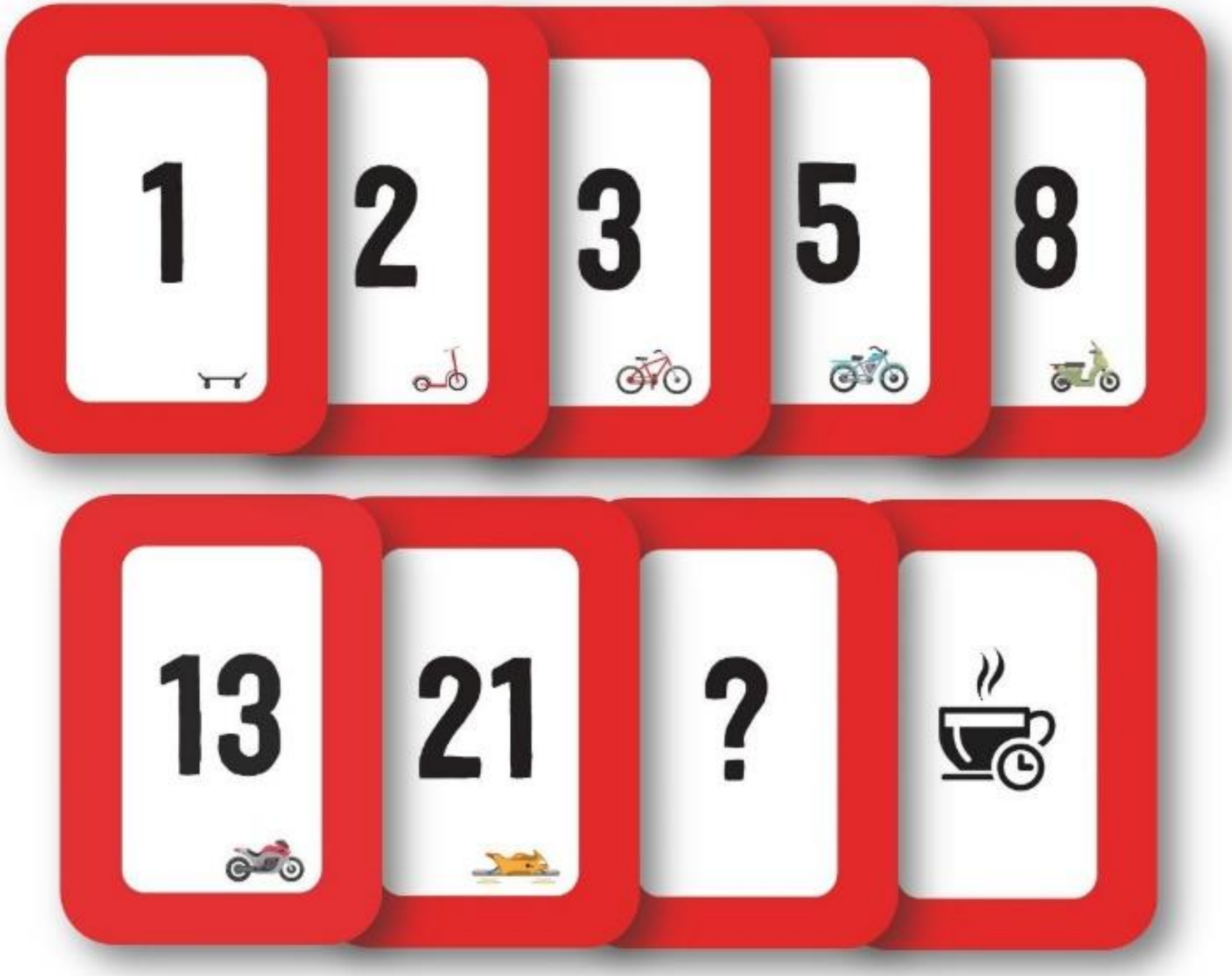
how relates A to B? A to C? B to C?

☒ **<u>Absolute estimation:</u>**

how big is A? B? C?



$$story\ points = f(size, risk, uncertainty, complexity)$$



1 2 3 5 8

13 21 ?

Why Fibonacci?

1, 2, 3, 5, 8, 13, 21…

Relative estimation is quicker and gives better results

Why are story points better than hours?

But you said it would take **forty hours** and now it has been two weeks!!

Alternative to numeric story sizes

T-shirt sizes: S, M, L, XL

Even simpler: small, big and unknown

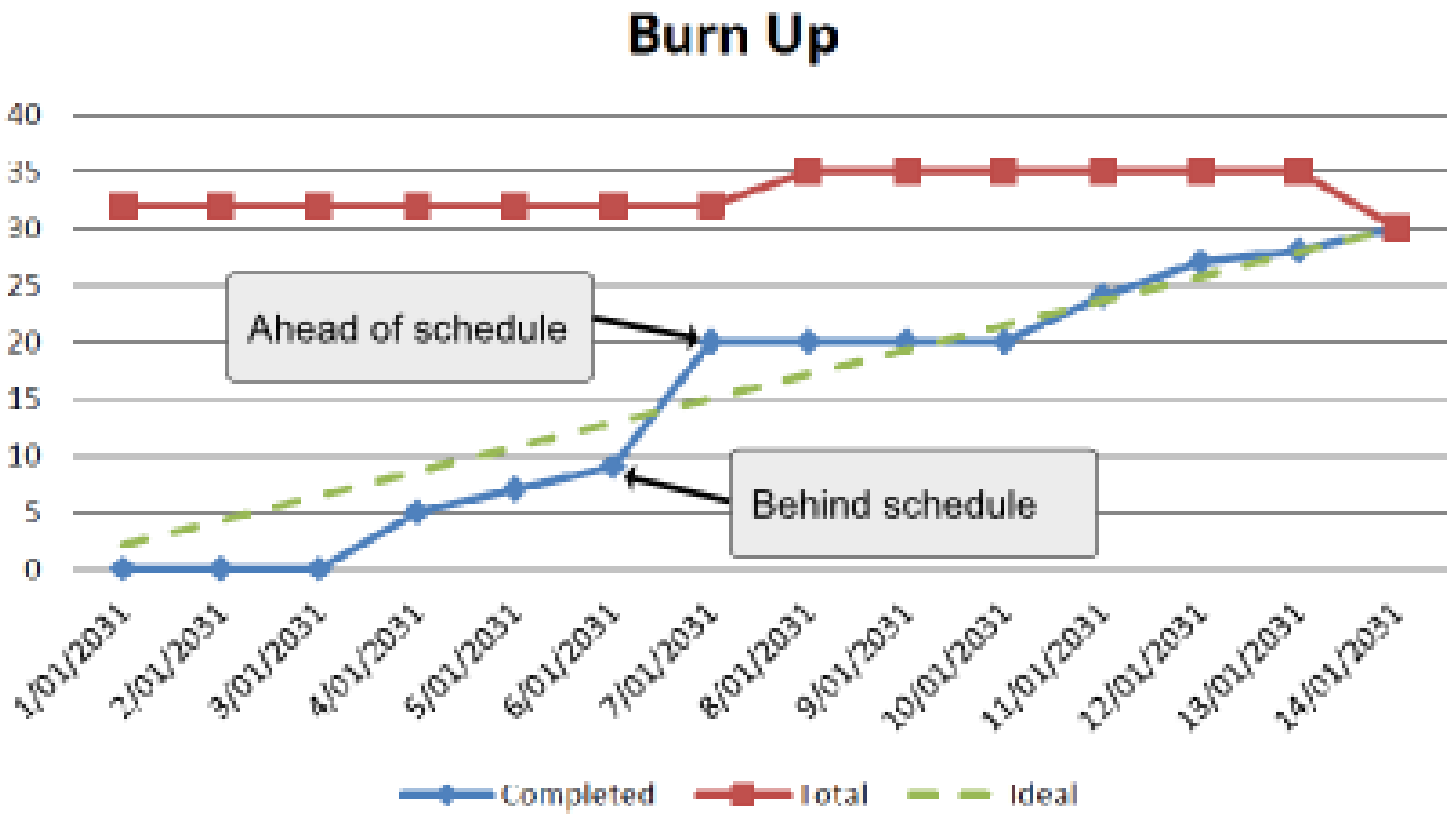
The purpose of estimation

To make people talk

How to know how you are doing?

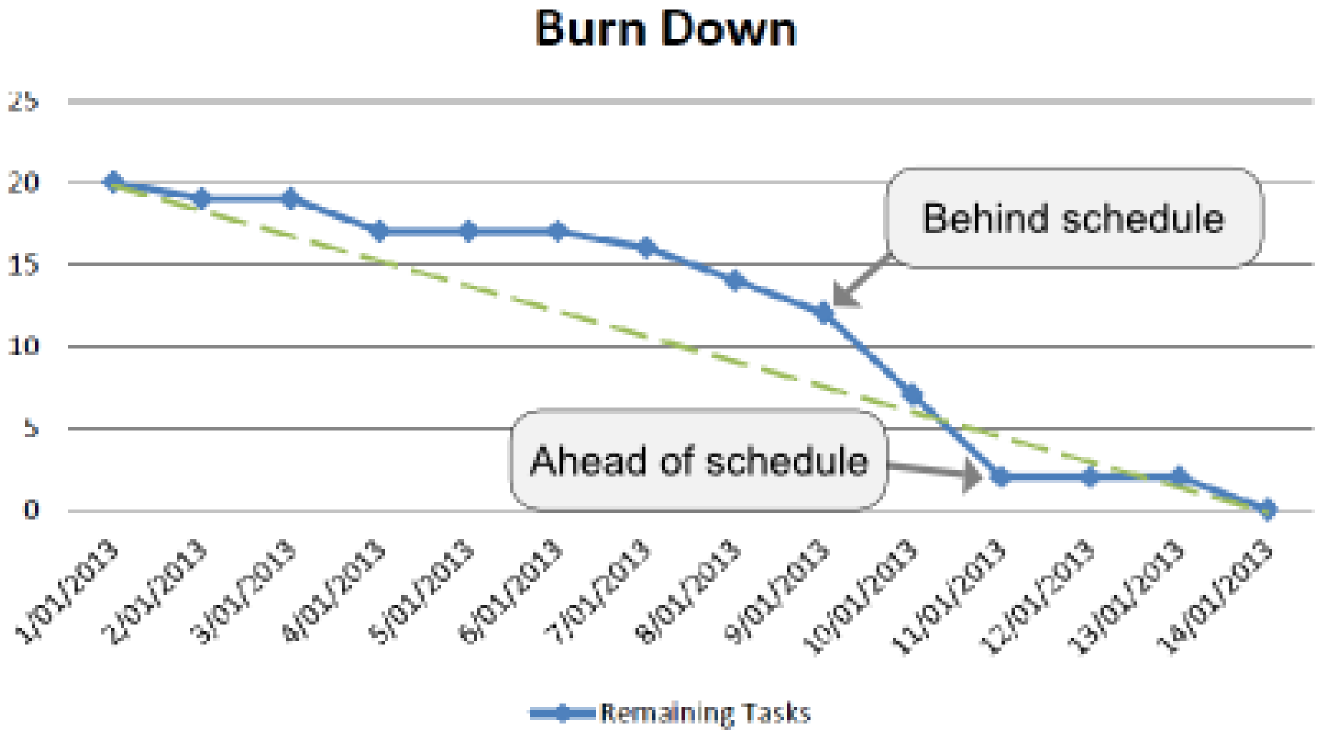
Burn up and burn down charts

Burn up chart – shows completed work



[What is a burn up chart? (clariostechnology.com)](https://clariostechnology.com)

Burn down chart – shows remaining work

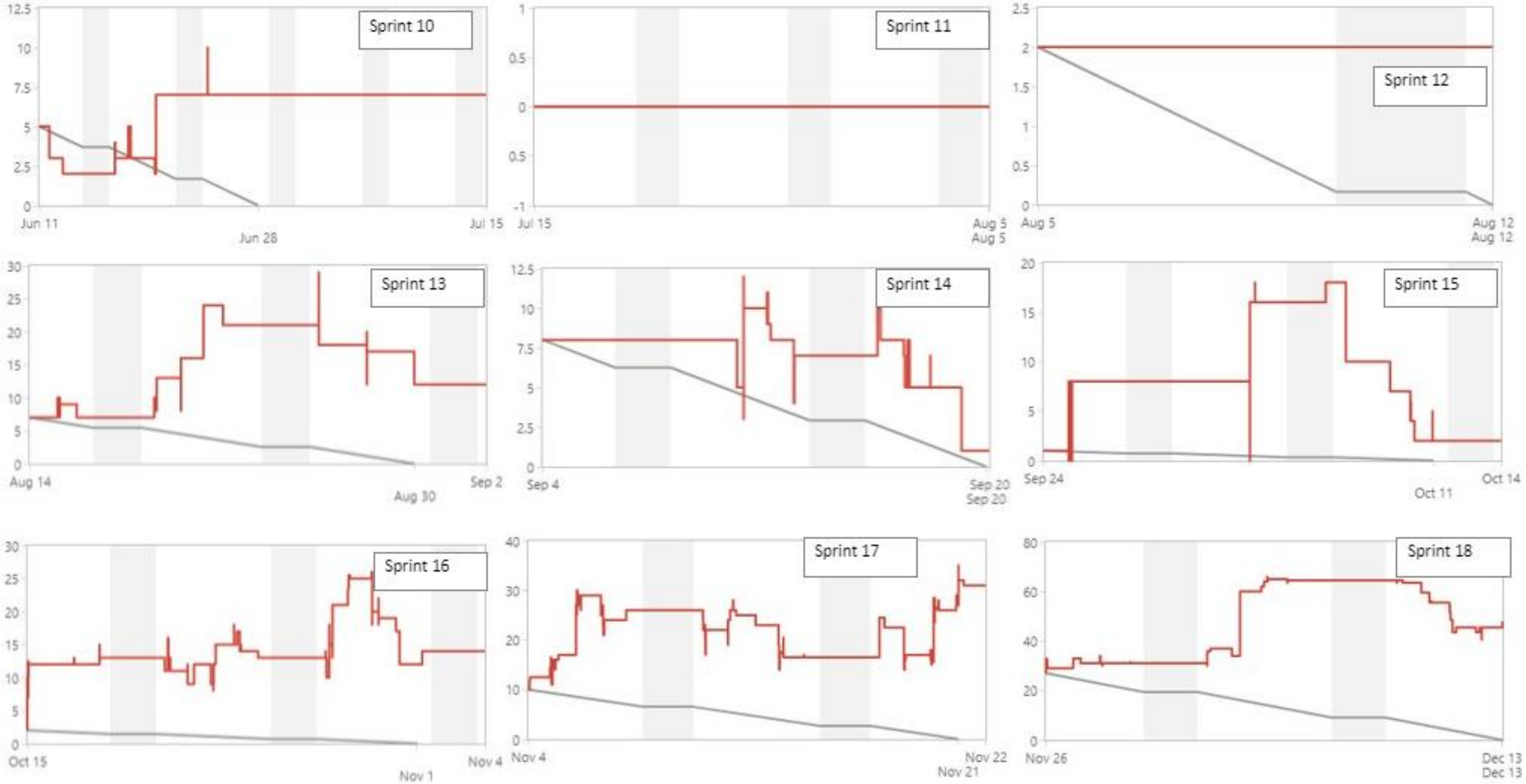


Behind schedule

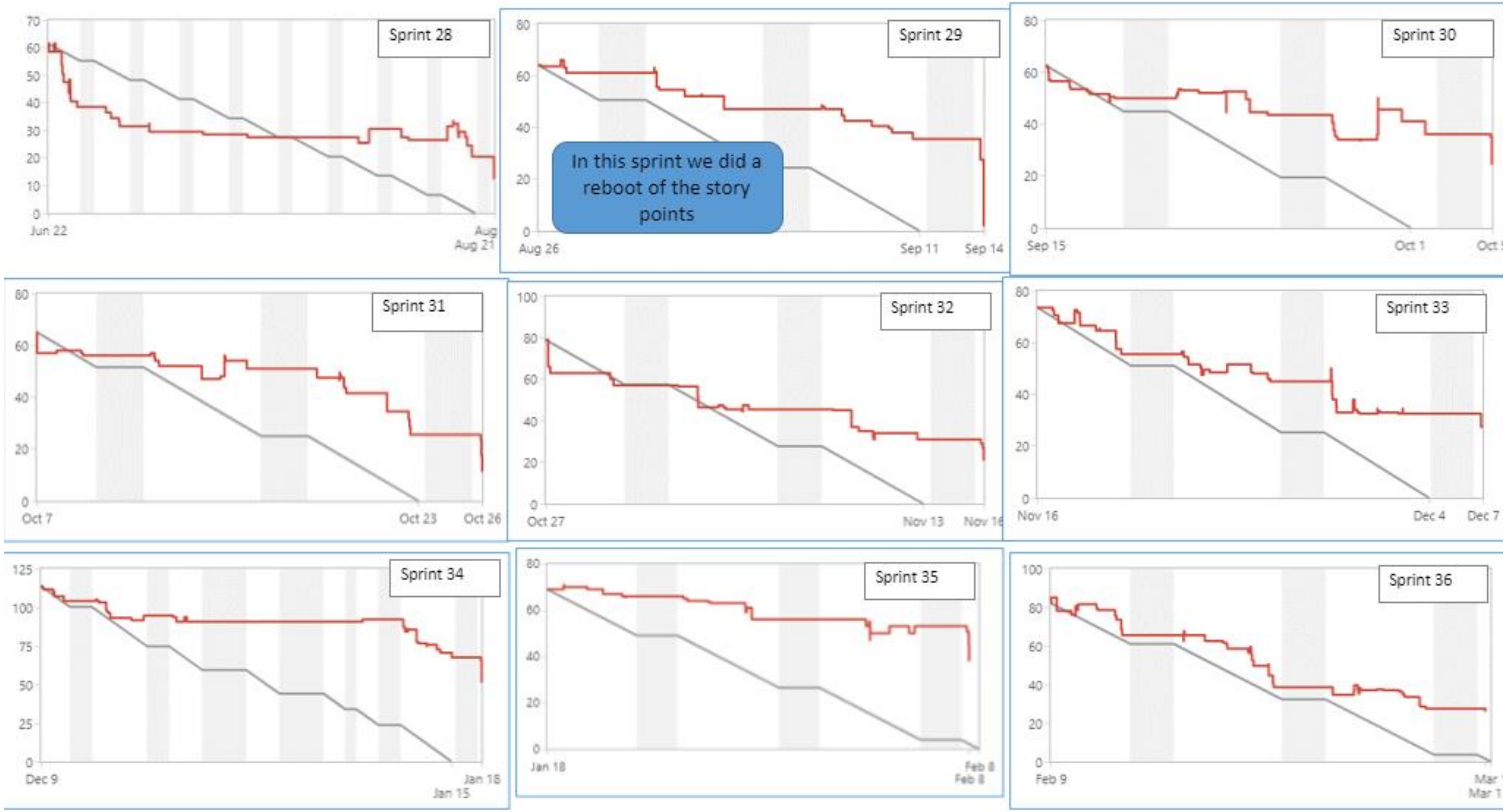
Ahead of schedule

[What is a burndown chart? (clariostechnology.com)](clariostechnology.com)

These burndowns mean nothing



These are not perfect but much better than before



In this sprint we did a reboot of the story points

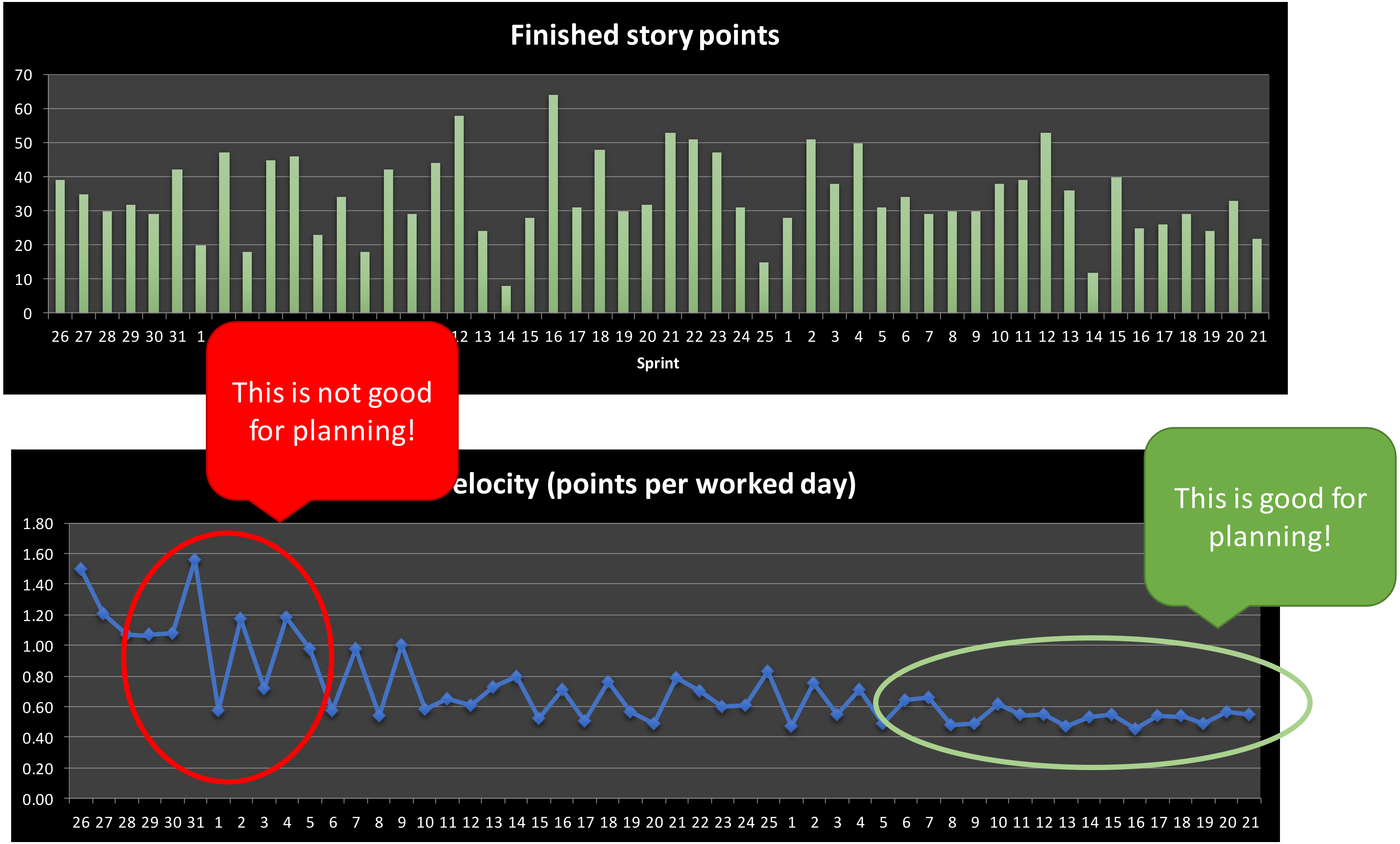
Velocity

The velocity of a team is based on their past performance

How much was ”Done”?

Note! Velocity is not a measurement of effectiveness, it's just the team's own metric to use for planning

!



This is not good for planning!

This is good for planning!

“Building more software faster is always a good idea. But, it’s never the solution.”

Jeff Patton

This is the correct priority

1) Do the right things
2) Make improvements

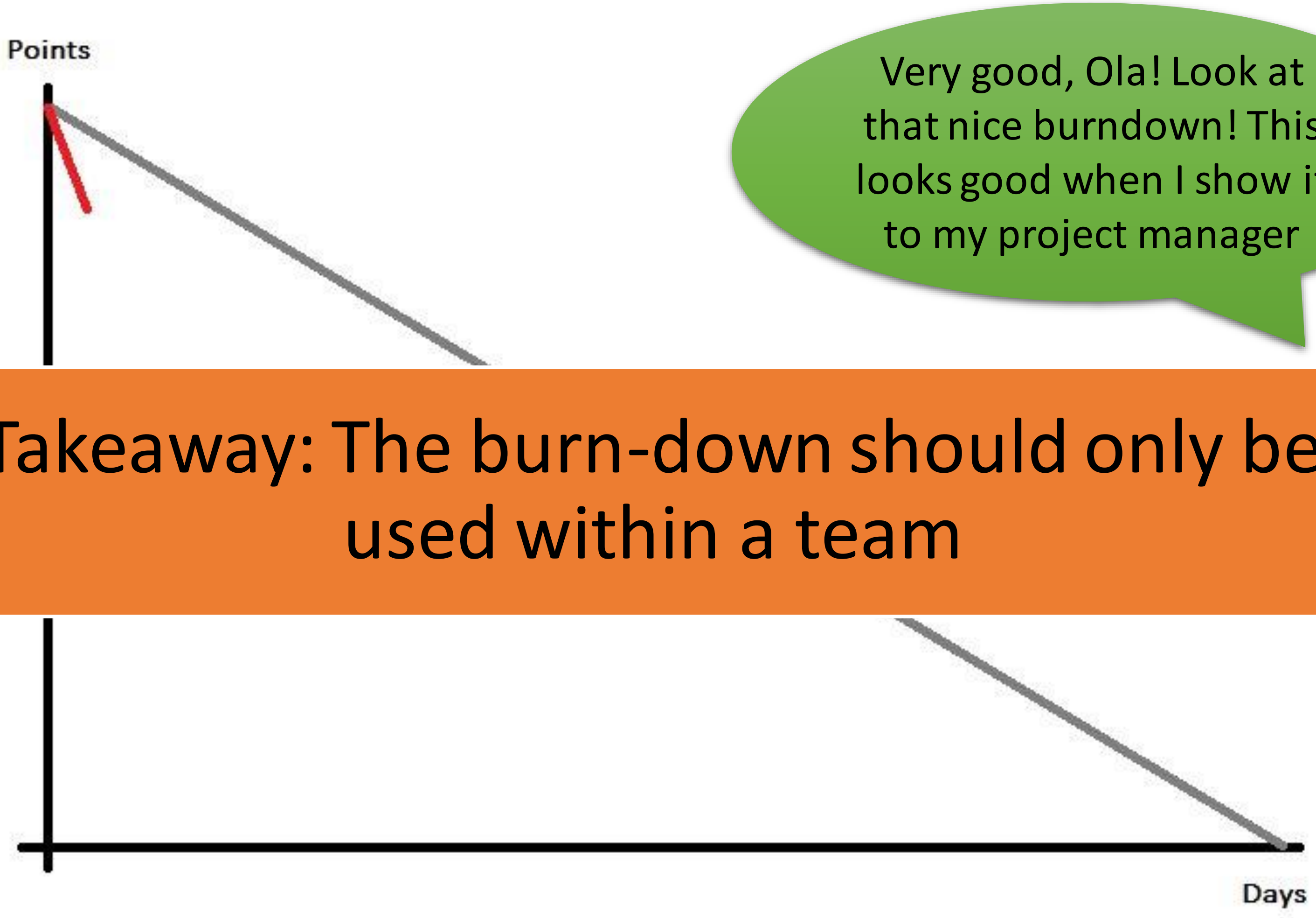
Don't:

Compare velocity between teams

Make story points to a game

!

Story: Cheating



Very good, Ola! Look at that nice burndown! This looks good when I show it to my project manager

Takeaway: The burn-down should only be used within a team

The “Scrum Master”

Other metrics

Metrics

Choose 1-3 metrics that help you quickly assess product health and detect if your improvement efforts are working

Example 1: For products like Twitter that might be the number of tweets per week, or average time spent reading tweets per day

Example 2: For Amazon it might be number of items in your shopping cart, or number of customers writing reviews for purchased items

KPI – Key Performance Indicators

A key performance indicator is a metric that you can pay attention to that lets you quickly see **IF YOUR PRODUCT IS WORKING AS EXPECTED** in the market

Look at the **TRENDS**: are you improving, deteriorating, or flat-lining?

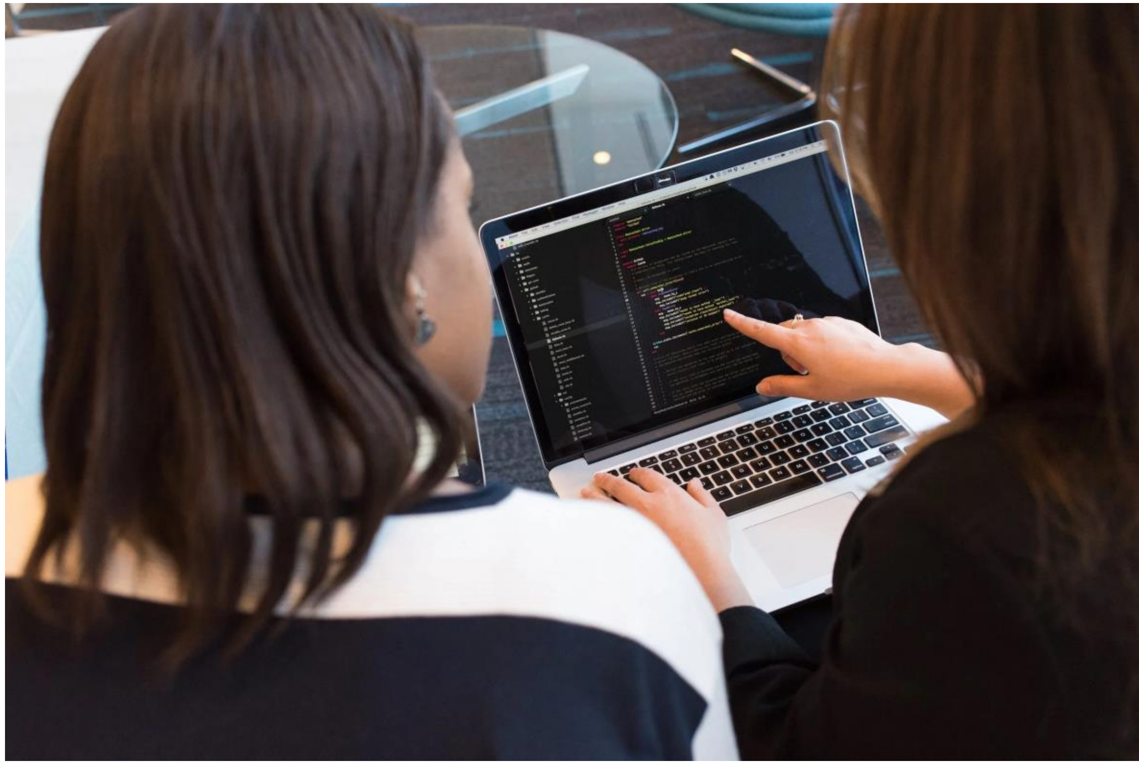
How do they **COMPARE** to what you expected? To last month? To last year?

Story: the tester and the developer

The tester was rewarded for how many bugs he/she reported

The developer was rewarded for how many bugs he/she fixed

What do you think happened?



Takeaway: You get what you measure

A more academic approach

We believe that __________________ (hypothesis)

To verify that we will _____________________ (do some test)
and measure ___________________ (some parameters)

We are right if __________________ (this criteria is met)

From Strategyzer

Example

We believe that ***more people will buy IKEA closets on the website if it was easier to navigate***

To verify that we will ***present in an attractive way which closets and additional items that are popular to combine*** **and measure** ***sales figures for closets***

We are right if ***sales figures go up 50 % in the next three months compared to last year at the same time***

DevOps metrics

Deployment frequency: how often does your organization deploy code to production or release it to end users?

State of DevOps report 2022

DevOps metrics

Lead time for changes: how long does it take to go from code committed to code successfully running in production?

State of DevOps report 2022

DevOps metrics

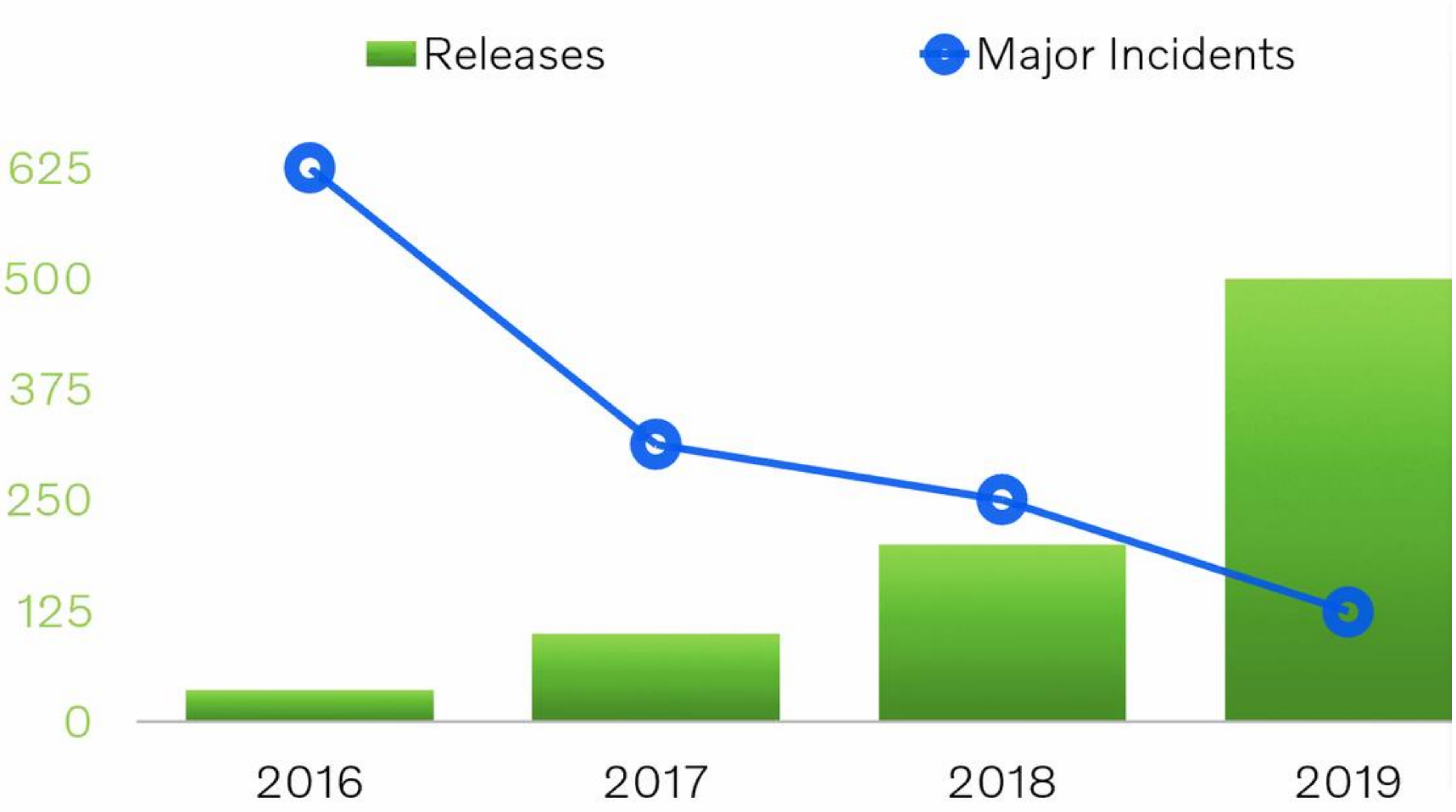
Time to restore service: how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g, unplanned outage or service impairment)?

State of DevOps report 2022

DevOps metrics

Change failure rate: what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, patch)?

State of DevOps report 2022



🟩 Releases
● Major Incidents
625
500
375
250
125
0
2016
2017
2018
2019

AARRR – Pirate metrics

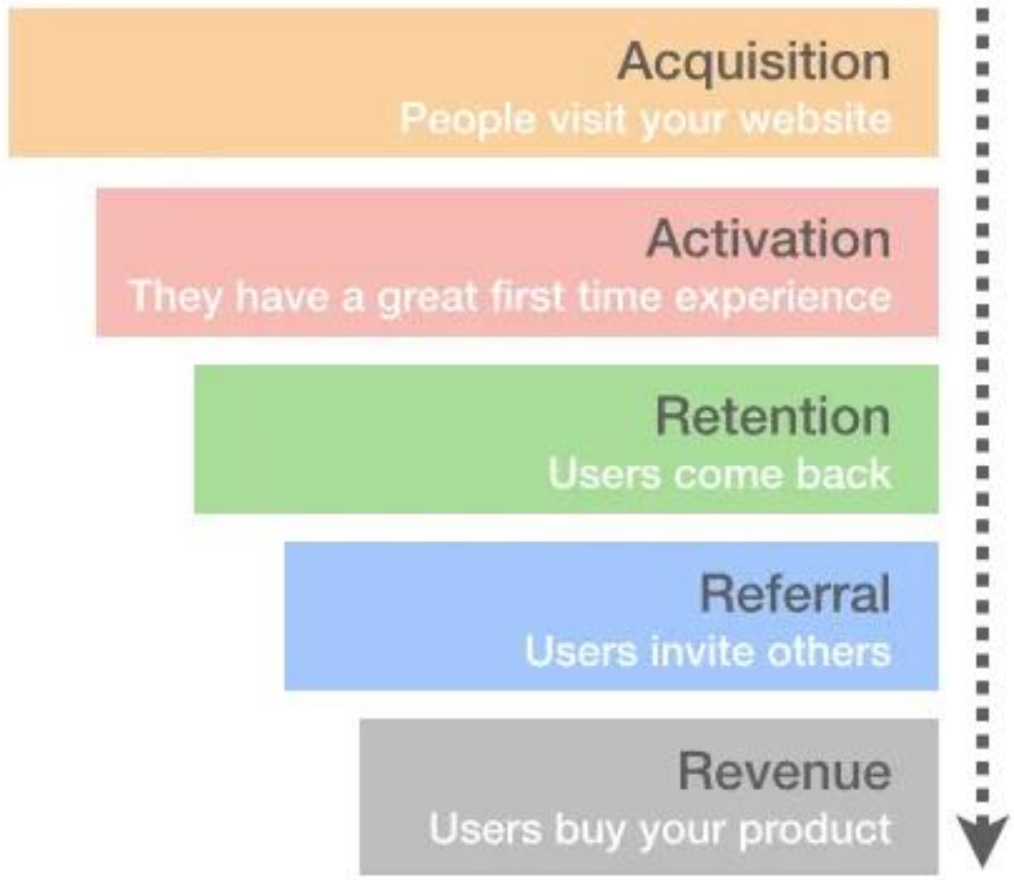
Acquisition – Opened the app, time spent, clicked around etc.

Activation – Create an account, visiting different sectors of the app, etc.

Retention – Are users coming back? Are they engaged?

Referral – Do users recommend us? Leave a review, unlock sections etc.

Revenue – Spend per customer, using premium features etc.



Acquisition
People visit your website

Activation
They have a great first time experience

Retention
Users come back

Referral
Users invite others

Revenue
Users buy your product

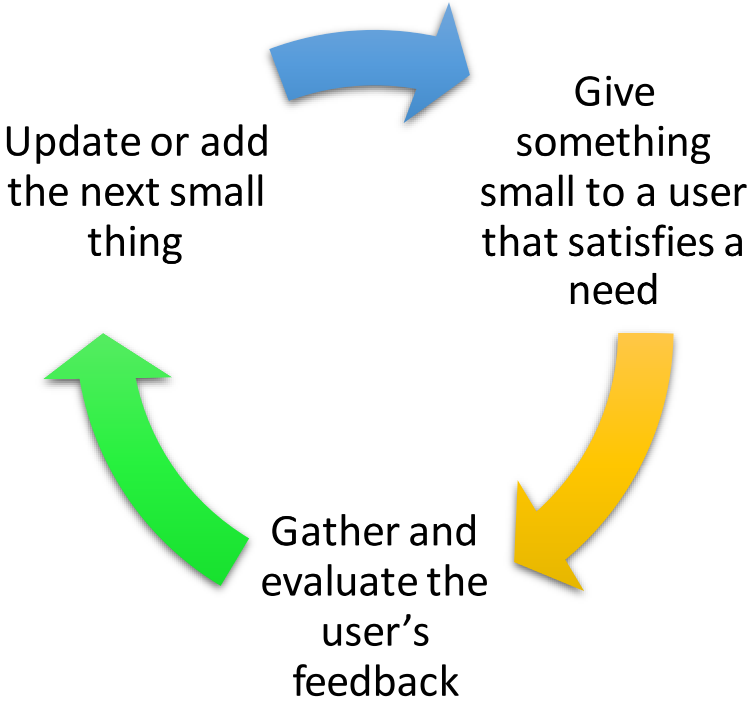
[What Is AARRR? Pirate Metrics Defined. | Built In]

[An introduction to the AARRR framework | by Gino Arendsz | Medium]

Ending

Key takeaways, 1 of 3

The expectation in agile is to deliver **SMALL THINGS OFTEN** and to **CONTINUOUSLY LEARN** what the users need and what we will do next to meet their needs



Give something small to a user that satisfies a need

Gather and evaluate the user’s feedback

Update or add the next small thing

Key takeaways, 2 of 3

Working in a good team is very rewarding, but it takes time and hard work to be a good team

- A **social contract** helps. Create one in your team this week!

Key takeaways, 3 of 3

Agile estimation, planning and working with metrics are difficult, but give it a try

- **Tip 1:** Use relative estimation (i.e. comparing with things you have already done). Don't use hours
- **Tip 2:** Make sure to talk and agree about what's included in a task
- **Tip 3:** Use one or two metrics if you think it adds value to your process

Feedback time!

Go to menti.com on your smartphone or computer and enter the code on the screen

Thea Schukken
www.beeldinwerking.nl