

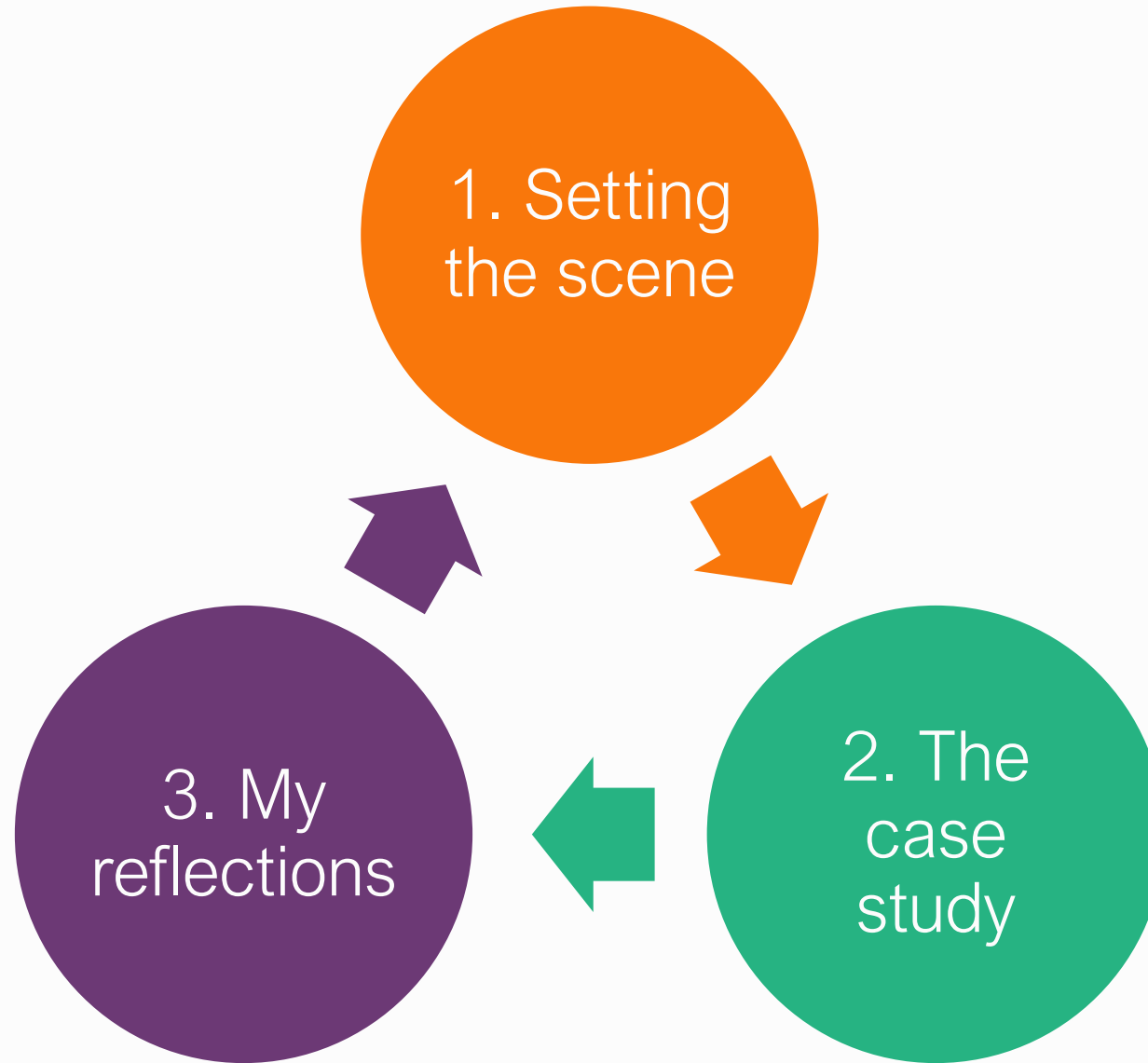
# Implementing Kanban in immature (and semi-hostile) teams

A case story



# My professional passions

- Kanban
  - Agile implementations and transformations
  - Agile coaching, consulting, and management sparring
  - Trainer: Kanban, Scrum, organizational agility, and agile leadership
- Volunteering, writing, speaking:*
- SME on Project Management Institutes' agile initiatives and exam writing workshops
  - Author
  - Public speaker
  - Blogger
  - Podcaster (Den Agile Agenda)



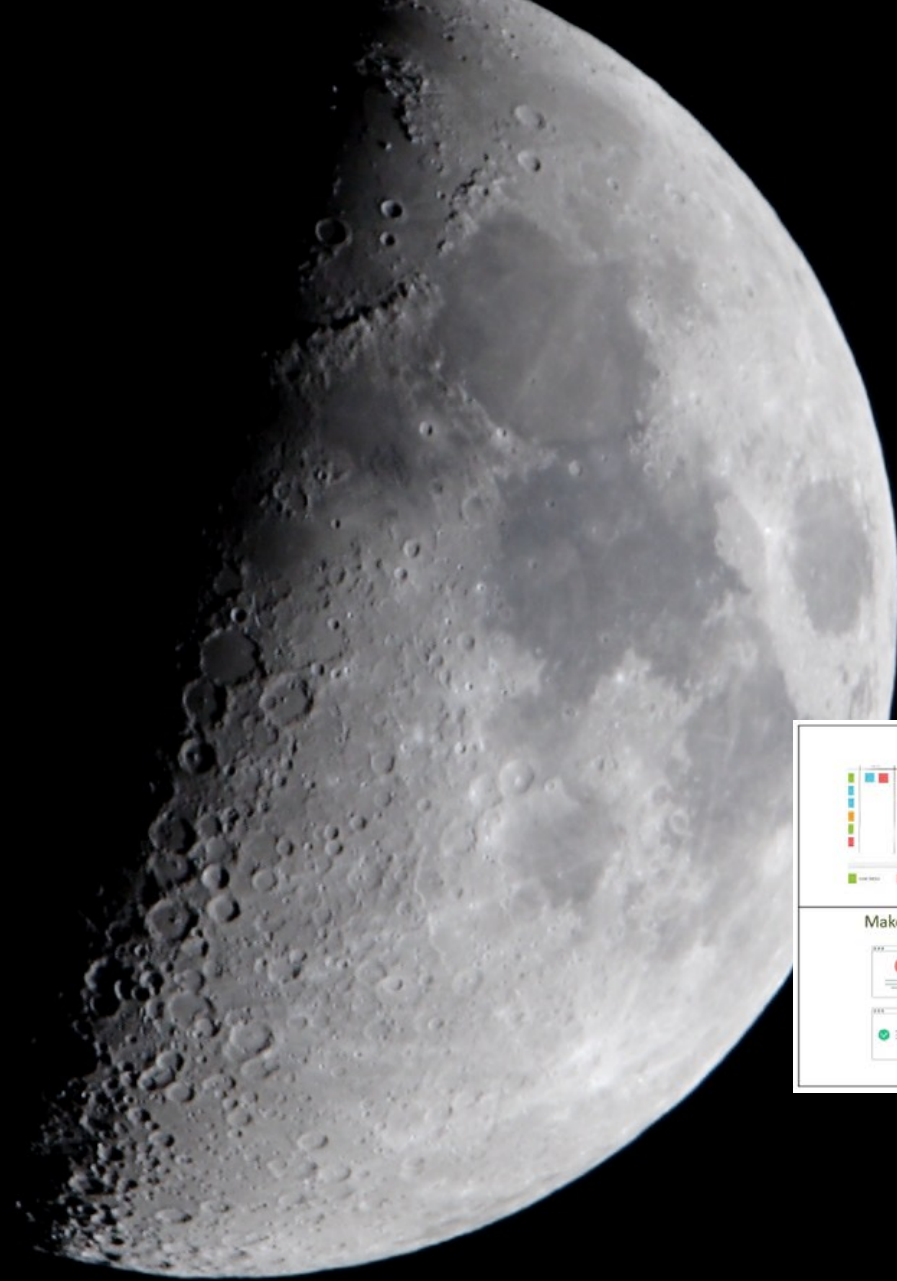
The difficult stuff is  
invisible / hidden



All the easy stuff is  
visible / transparent

The difficult stuff is invisible / hidden

All the easy stuff is visible / transparent



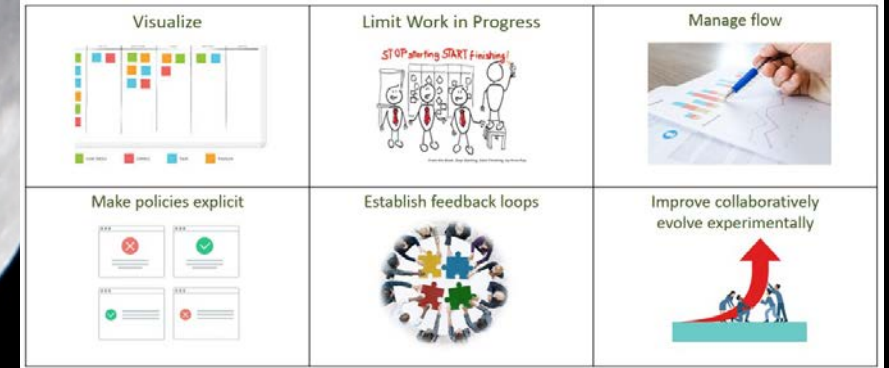
<p>Visualize</p>	<p>Limit Work in Progress</p>	<p>Manage flow</p>
<p>Make policies explicit</p>	<p>Establish feedback loops</p>	<p>Improve collaboratively evolve experimentally</p>

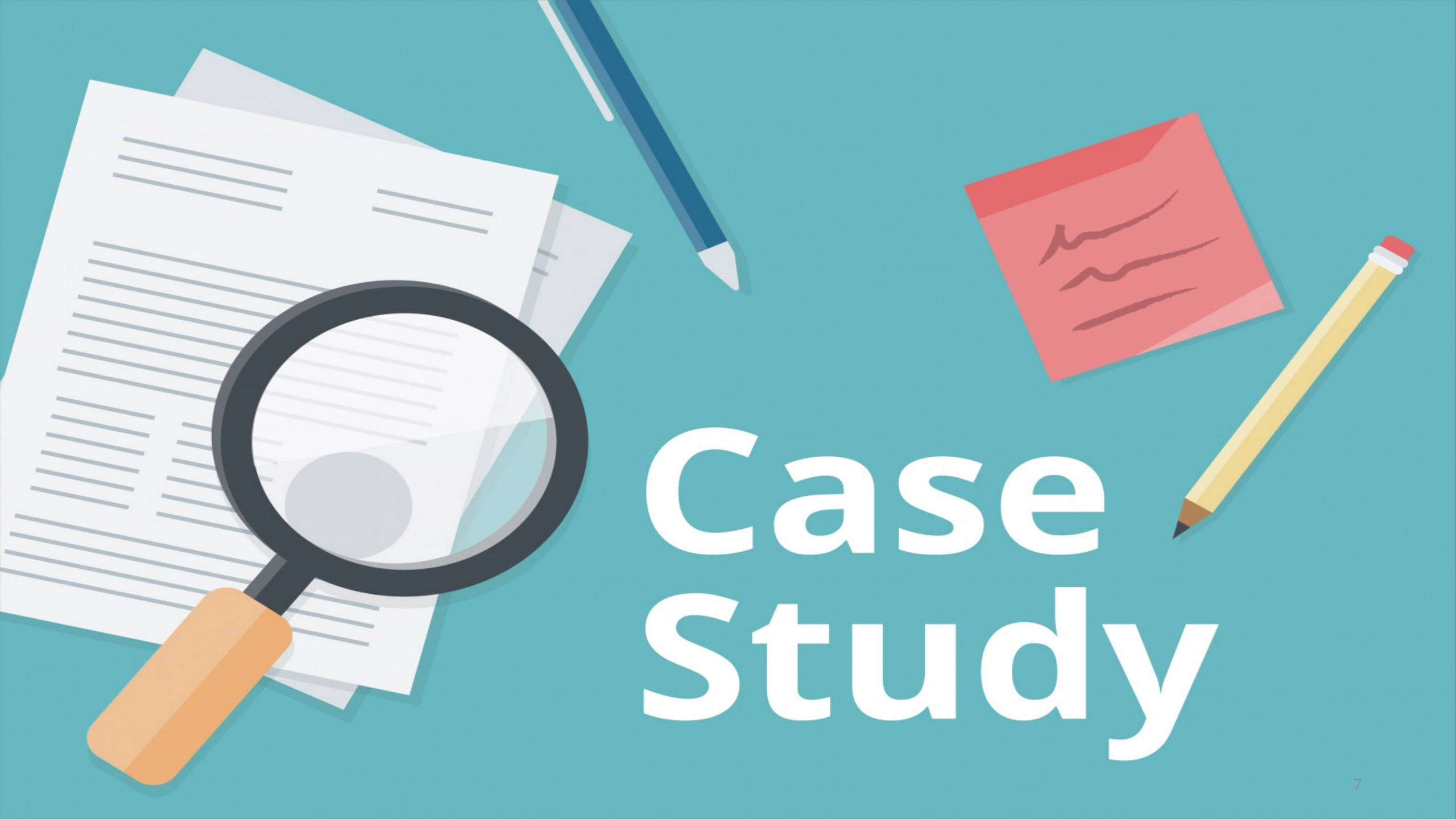


The difficult stuff is  
invisible / hidden

- Silo mentality
- Organizational culture
- Politics
- Business involvement
- How does new requirements arrive?
- Error tolerance
- Change readiness
- Habitual thinking
- Management style
- X-team dependencies
- (Governance)

All the easy stuff is  
visible / transparent



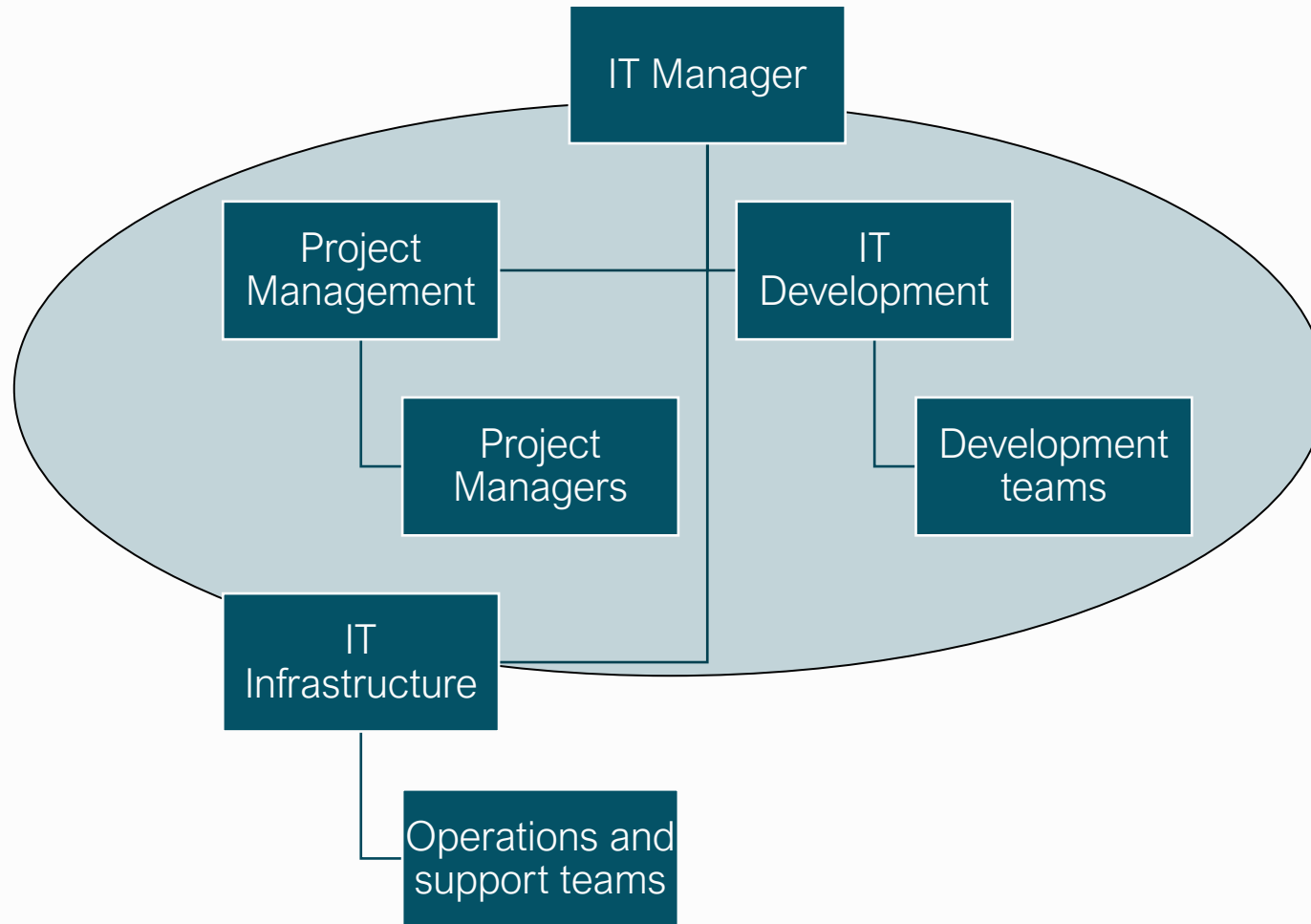


# Case Study

# A Kanban implementation in the IT department of a big university in Denmark



# The high-level IT organization + our scope

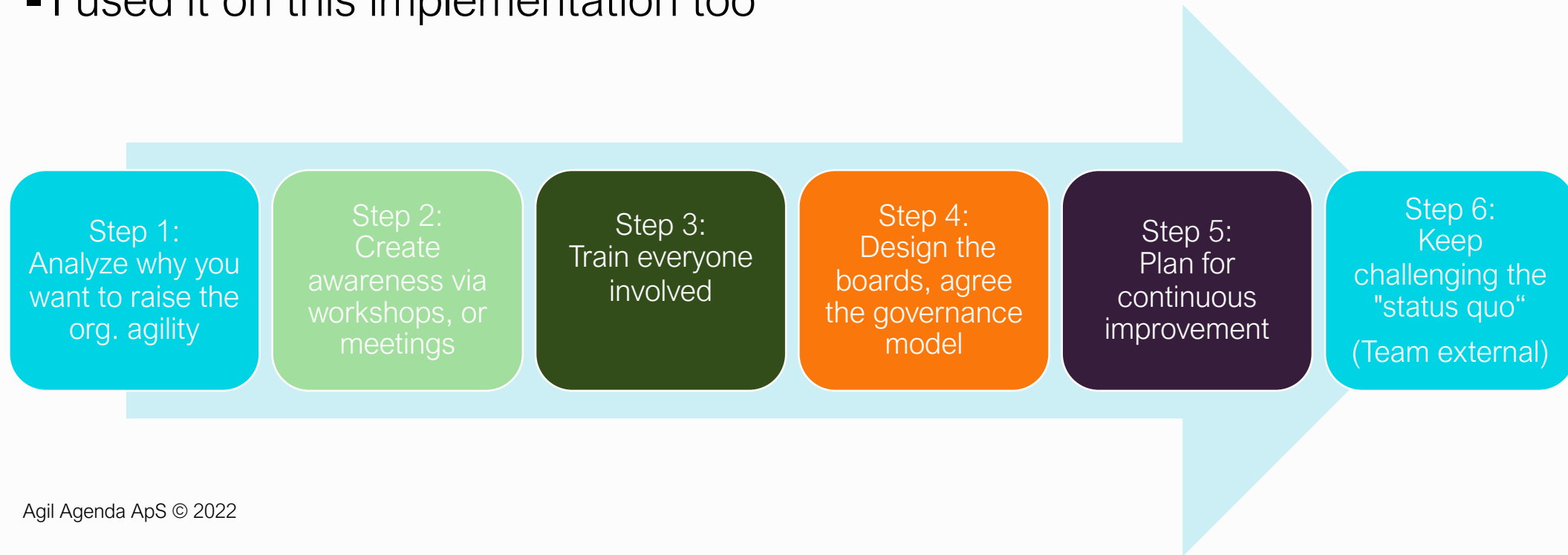


The IT Department delivers services to nearly 30.000 full-time and part-time students, professors/teachers, library, admin. personnel etc.

There are cross-team dependencies between all teams in the IT organization

# My general approach to agile implementations

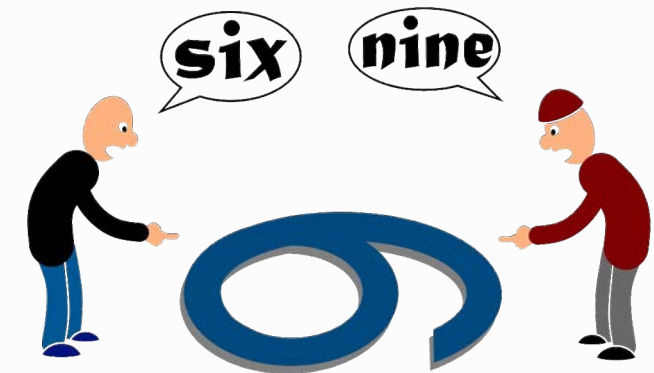
- The model is based on extensive experience with customers
- I used it on this implementation too



Agil Agenda ApS © 2022

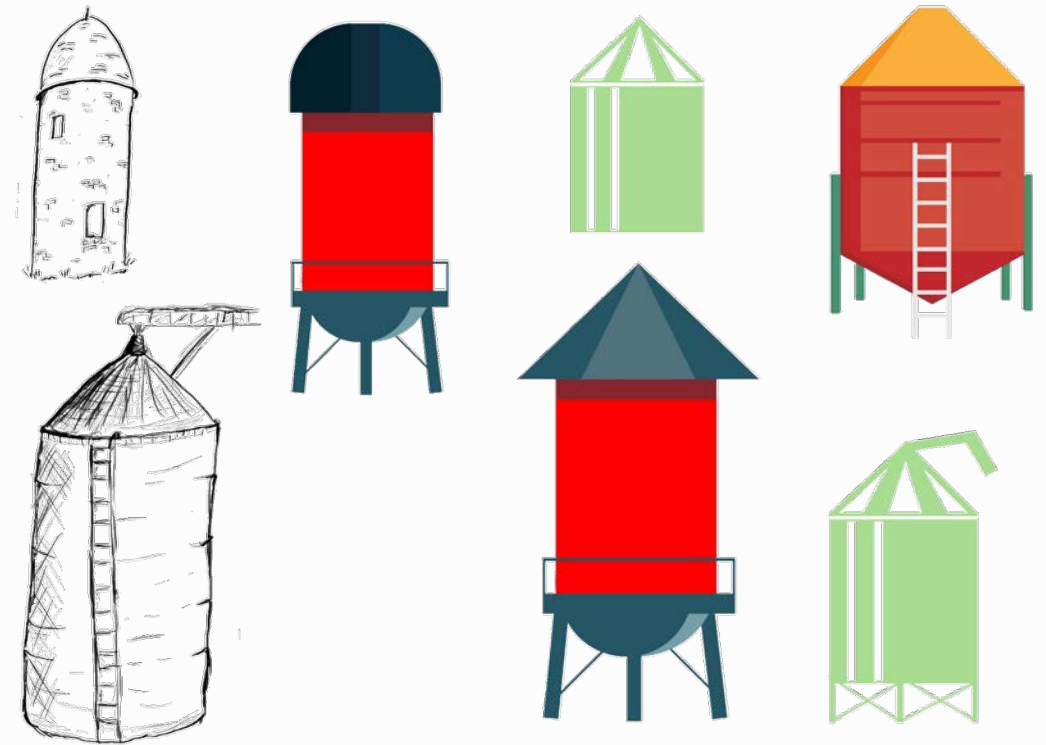
# Step 1: What did they struggle with, and why agile?

- IT development teams and Project Managers were speaking 2 different languages. They did not understand each other
- Communication and collaboration was poor. Finger pointing and blame games were normal
- Projects were very often late, and budgets were overspent
- Customer satisfaction was quite low, and so was employee satisfaction
- New wishes/requirements came from everywhere
- No predictability
- Not much transparency in who worked with what and why
- (Inefficient) “Management by Excel” for resource allocation



# They worked in silos – not 2 silos were alike

- Silo mentality
  - Different project silos
  - Different team silos
  - Departmental silos
- Sub-optimization
- No interest in other people's problems
- Lack of value-chain thinking
- They worked “Scrumish” (as they said) but each team did it in their own way



# Step 2: The awareness workshops

We used questionnaires, team exercises and did lightweight Scrum and Kanban training

They could answer yes, no, or I don't know

The questionnaire (20 questions re. agile good practice). A few sample questions:

- Do you and your team always work according to clearly defined requirements / Product Backlog?
- Do you break down activities to 1-5 days' duration?
- Do you know who your client, your Product Owner or end-users are?
- Do you deliver the quality that you would like to
- Do you "work" on more than 3 activities at the same time?
- Is the communication between you/your team and the business efficient?
- Are you testing immediately after development
- Do you ask your PO, PM or end-user if you are in doubt of what they want?
- Do you always have a Scrum Master
- Are you aware of critical deadlines

Spørgeskema ved workshop om brugen af agile

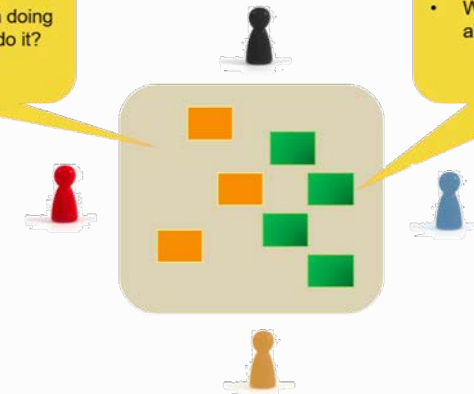
Team: \_\_\_\_\_

	Ja	Nej	Ved ikke
1. Arbejder du altid ud fra klart definerede krav / en gennemarbejdet Product Backlog?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Sikrer du, at alle projektaktiviteter til 1-5 dages varighed?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Kender du altid de deadlines, der er vigtige i dit projekt eller for din projektleder?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Er du god til at overholde de aftalte deadlines?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Ved du altid, hvem din kunde, Product Owner, eller slutbrugere er?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Kommunikerer du og din projektleder / Product Owner ofte nok?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Holder de estimater, som du giver på dine aktiviteter, som regel? (D.v.s. +/- 5-10%)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Melder du tilbage, hvis du kan se, at dine estimater ikke holder?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Melder du tilbage, hvis du kan se, at du bliver forsinket?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Føler du, at du har mulighed for at levere den kvalitet, du gerne vil?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. Spørger du din projektleder, Product Owner eller kunde, hvis du er i tvivl om, hvad de forventer, at du skal udvikle?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Har I altid en Scrum Master?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. Får du testet dine ting straks efter udvikling?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. Dokumenterer du det, du har udviklet med det samme?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. Er du og dit team enige om, hvordan I skal gribe jeres opgaver an? (Løser I ens opgaver på den samme måde?)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16. Har I standardskabeloner for tilbagevendende aktiviteter?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17. Bruger du de skabeloner, som CBS evt. stiller til rådighed?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18. Har du mere end 3 ting i gang på samme tid?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19. Får du lov til at fokusere på at gøre én ting færdig ad gangen?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20. Multitasker du mere, end du har lyst til og ved er effektivt?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Discovering the sources of dissatisfaction

- What are you dissatisfied with?
- What makes you frustrated?
- What is stopping you from doing the work like you want to do it?
- When do conflicts arise?

- Why are your customers dissatisfied?
- What do you think people think about the work you do?





## Step 3: Train everyone involved

- That's what we did 😊
- There was high management buy-in at this point in time
- We had the budget
- They wanted to get off to a good start



# Step 4: Designing the boards and the Kanban system

- Our ambition was to build a Kanban system, where nothing would fall through the cracks
- Each team had some (not much) autonomy in designing their own Kanban board
- They agreed the team rules, e.g., Definition of Ready and Definition of Done
- But... the management team wanted to standardize in terms of visualization techniques
- ... and the management wanted the teams to measure and report on progress once a week

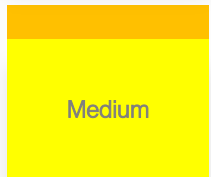
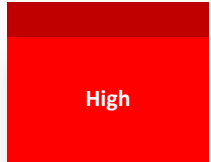


# Common board rules

## How to visualize and create pull signals

### "Stop starting – start finishing"

#### Class of Service



Write when and why the activity was blocked as well as the actions planned or already taken



We measure once a week



Blue magnet = internally blocked (= we can act to get the blocker removed)



Black magnet = externally blocked (= we have no options to act. We can only wait )



Red magnet = the activity is now urgent



Green magnet = the activity is done and can be pulled to next column (= a pull-signal)



Yellow magnet = watch out! (For any reason)



Definition of Ready = .....



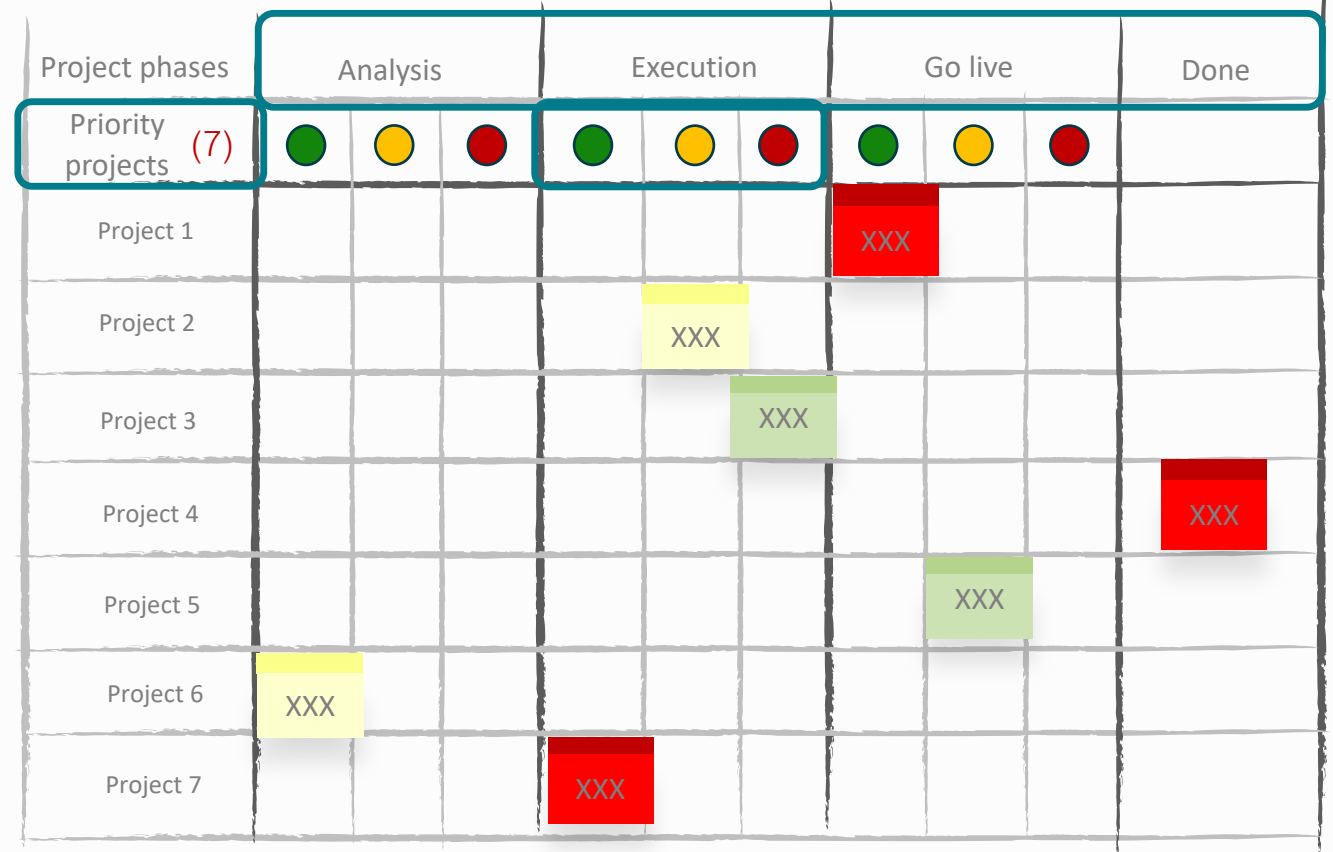
Definition of Done = .....

# Project Management Office (PMO) boards

Projects in the pipeline = upstream Kanban



Ongoing projects = downstream Kanban



Priority:

High

Medium

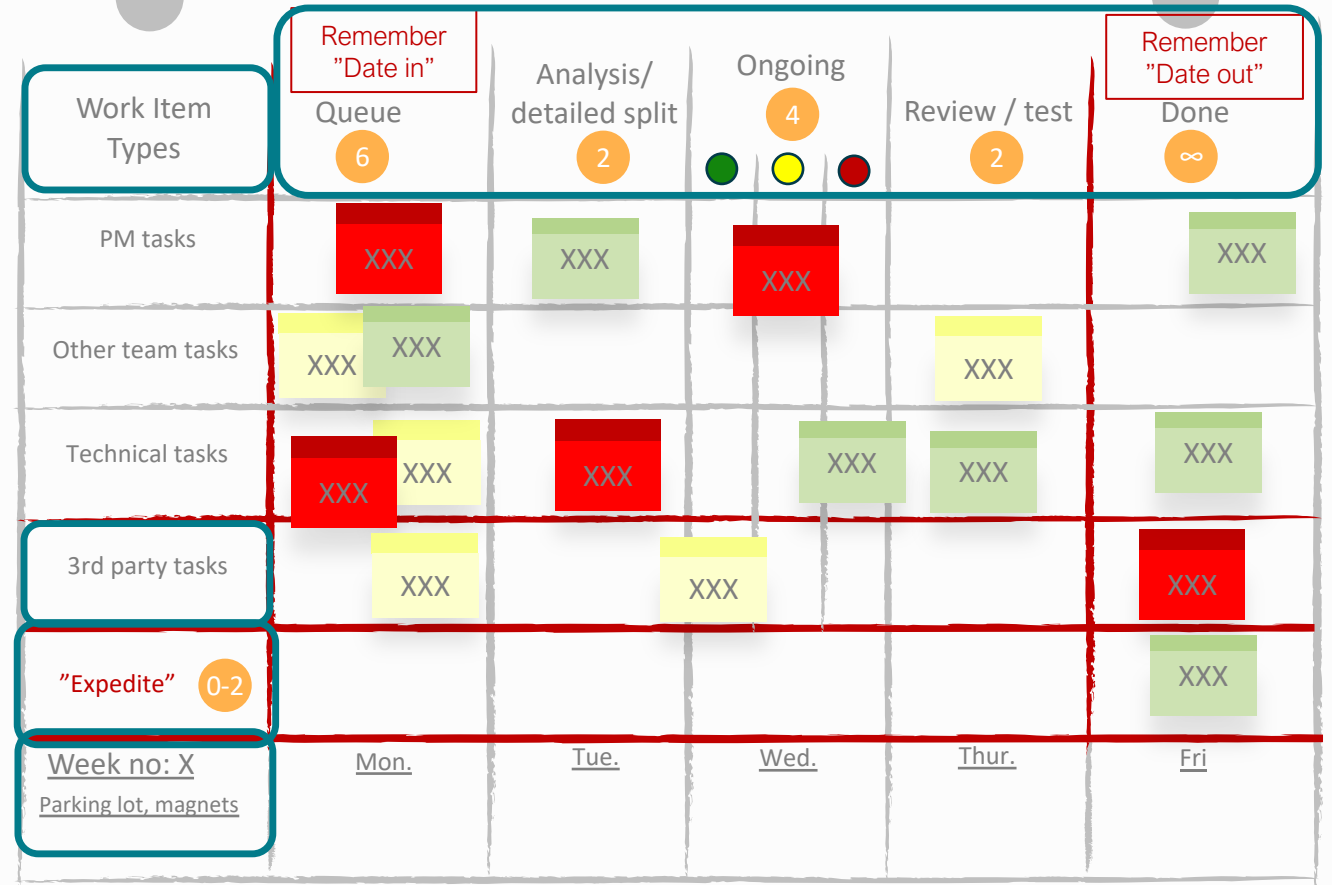
Low

# A Project kanban board

New requirements / ideas =  
**upstream** Kanban



Delivery board = **downstream** Kanban

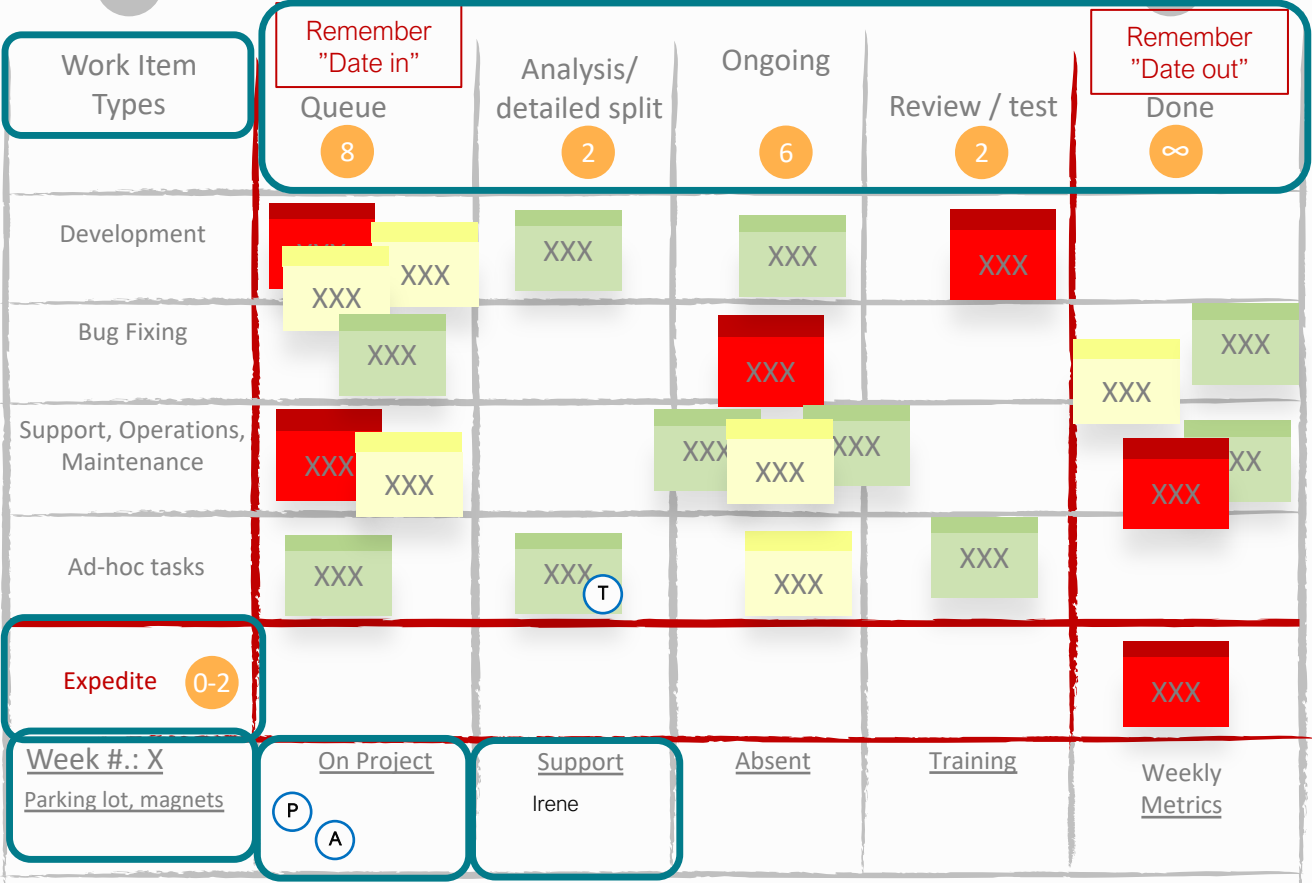
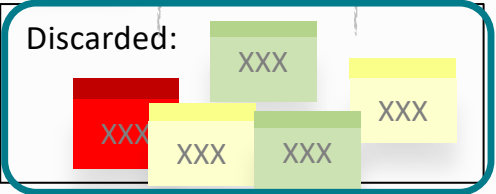
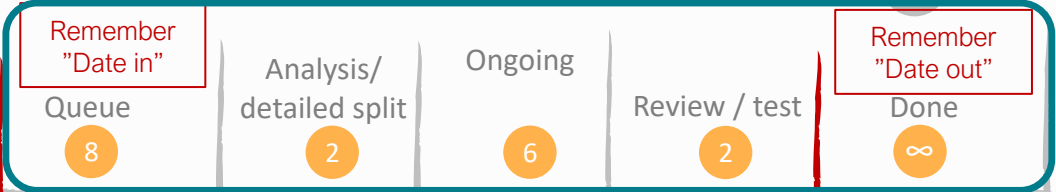
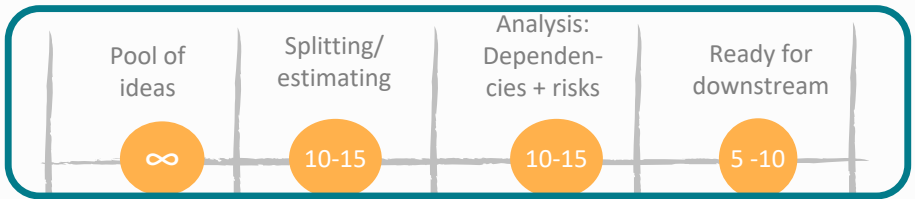




# A team kanban board (5 members)

New requirements / ideas =  
**upstream** Kanban

Delivery board = **downstream** Kanban



# Dependency Board

## Board rules

Depen-  
dencies

Only dependencies, that might result in broken deadlines, or will delay (project) teams in proceeding with their work, delays in deployment etc. Should be on the board

Urgent depen-  
dencies

Conflicts, e.g. 2 teams needing help from the same specialist at the same time. Our managers must prioritize the different activities

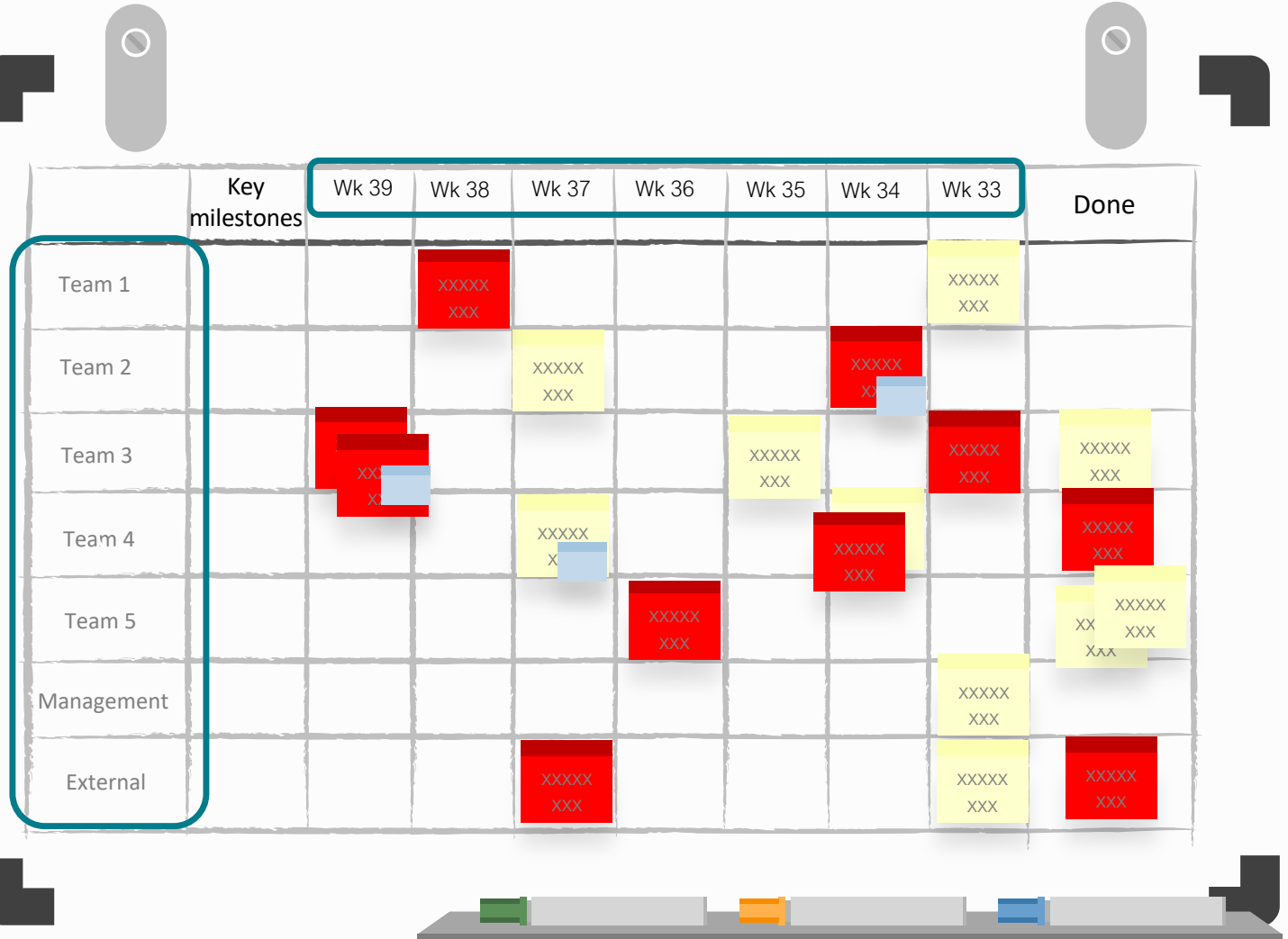
New  
item

In the column "Key Milestones", place a big PostIt with a short description of the activities in focus during the period on the board

The PostIts should show:

- When they were placed on the board
- A description of the activity that a particular team or person in the team must deliver
- The week or date on which the activity must be completed

Place a small blue  
Post-it on new  
yellow or red Post-its



**Easy to say – not so  
easy to do...**

## Step 5: Planning for continuous improvement (to do so, we had to measure to get a baseline)

- Starting up we decided to keep it simple. We measured:
  - System lead time
  - Throughput
- I had the ambition to measure...
  - Number of blockers
  - Quality: New errors arriving vs. number of old errors corrected in the period
- ... but this was not immediately accepted. Maybe later
- Instead, we agreed to discuss the use of the 6 general practices in team retrospectives together with lead time and throughput



## Step 6: Challenge the status quo & Plans for the longer term

Transforming an organization is a journey that will most likely stop or be only team-centric if you do not plan how to make it continue

This is in the pipeline at the university:

- Maturity assessments
- Common views on agile behavior to create a common agile culture
- Process improvements  
(Cross-team and Cross-departmental)





**Change is never easy, particularly  
not in immature teams**

**– My most important learnings**

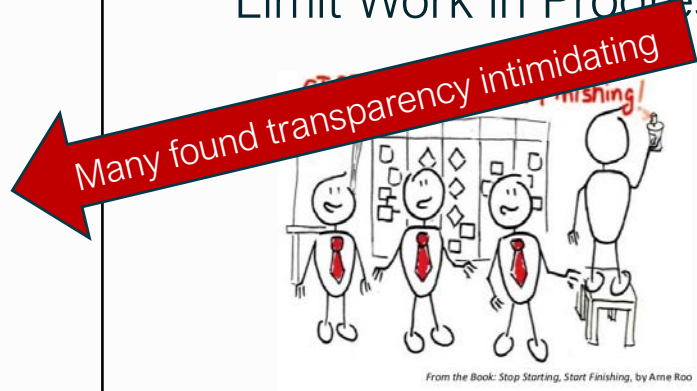
**Leadership!**  
**Leadership!!**  
**Leadership!!!**

# The 6 core practices – the difficulties

## Visualize



## Limit Work in Progress



## Manage flow



## Make policies explicit



## Establish feedback loops



## Improve collaboratively evolve experimentally



# Team values were all different...

- Don't underestimate the power of team values and tribal behavior
- Each team had their own set of values. They were very difficult to figure out
- The lack of a common vision made each team form their own
- Most teams were reluctant to give up their “enemy pictures”

My conclusion: Put more energy in defining “something” that all teams would want to fight for together

Recommendation: Ray Immelman: Great Boss – Dead Boss



# It was difficult to leave old estimation practices

The fact that task switching, wait time, blockers, dependencies etc. made estimates almost useless, and that lead time is more reliable, was simply too big a bite to swallow.



# Key take-aways



In change projects incl. Kanban implementations, it's the invisible things that make it difficult

Transformations can and should be planned. Set some goals up front, so you know what you are aiming at

Be realistic and pragmatic when you work with low maturity teams. Don't set the bar too high

One thing is the method another thing is people. Look out for tribal behavior. Deal with it, if you can

# Contact information

 annette@agilagenda.com

 www.agilagenda.com

 xvoto.wordpress.com

You can also reach me via LinkedIn, Twitter, and Facebook





