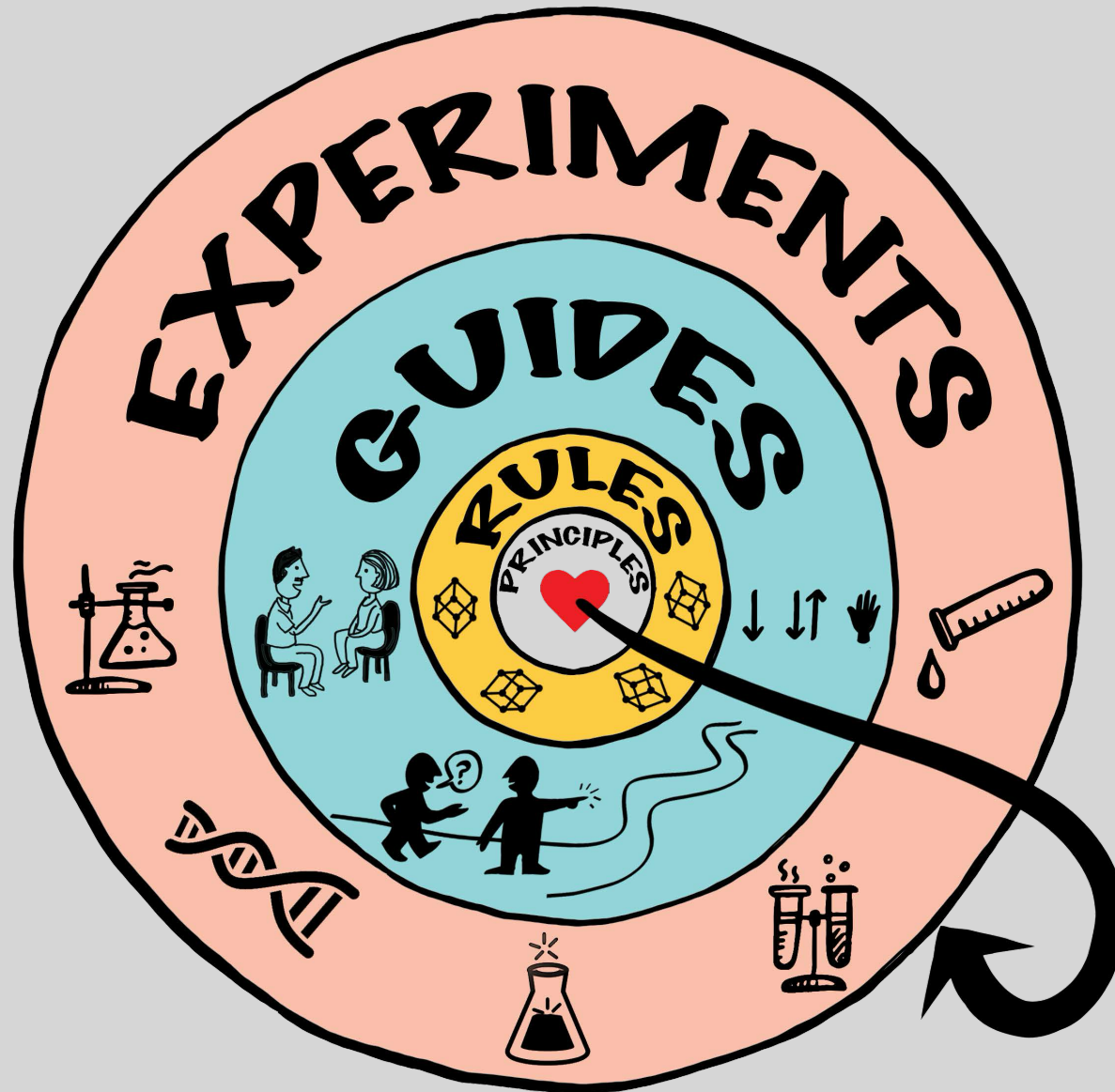


<eSS



Secret measurement of healthy team dynamics

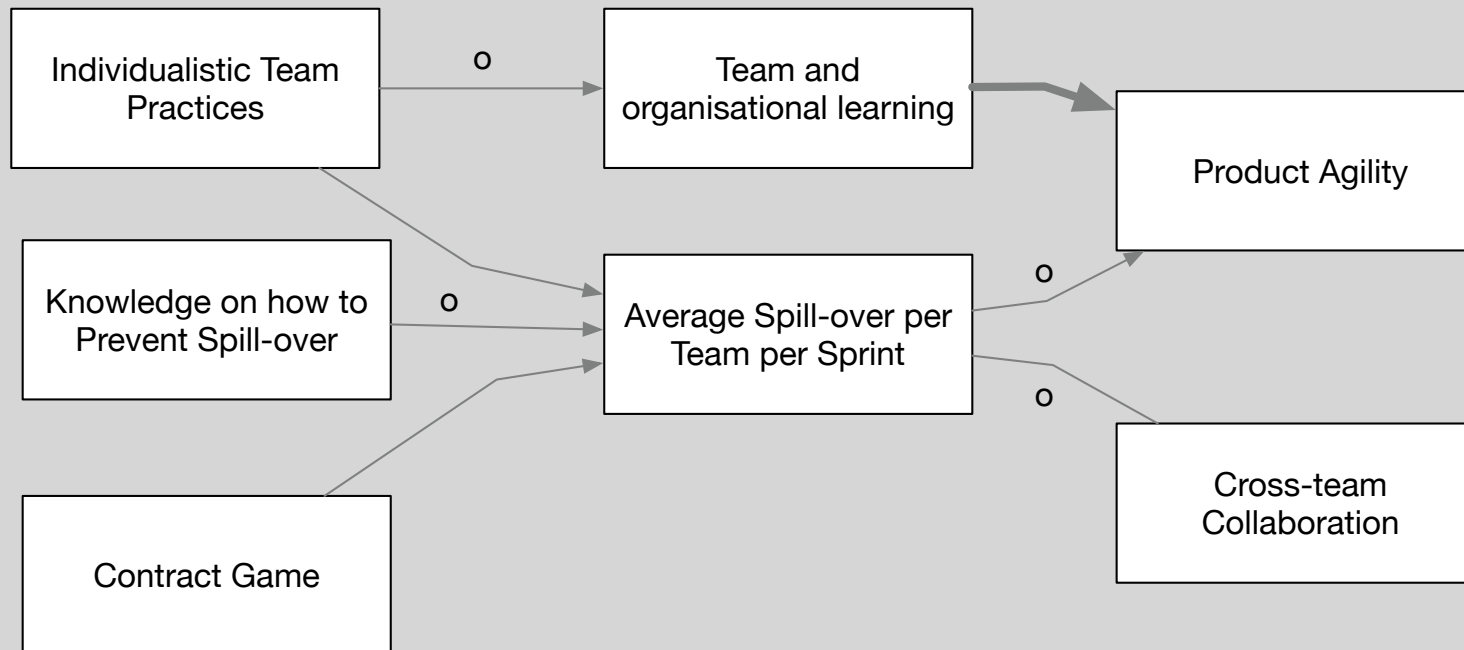
Large Scale Scrum



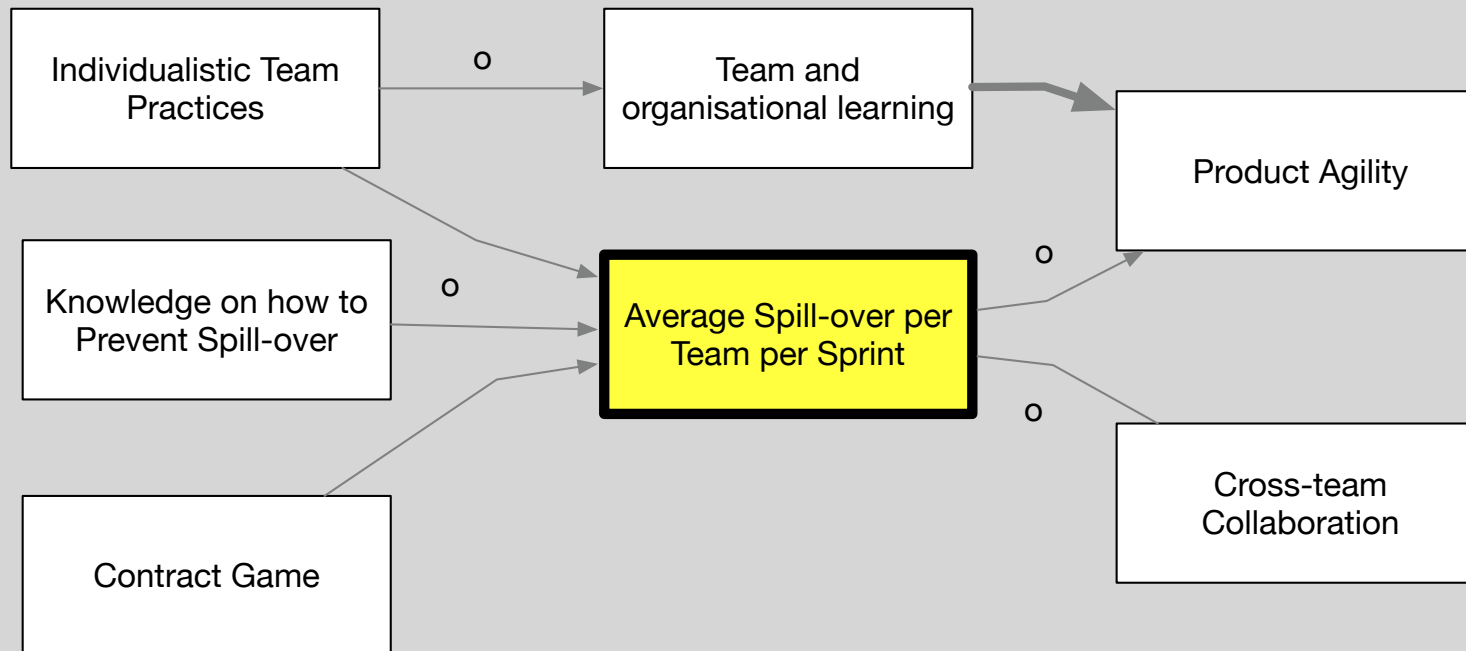


**Why are you
always so obsessed
with spill-over?**

Agenda



What is spill-over?





Definition Spill-over

Amount of items that are in progress at the end of the Sprint



Examples

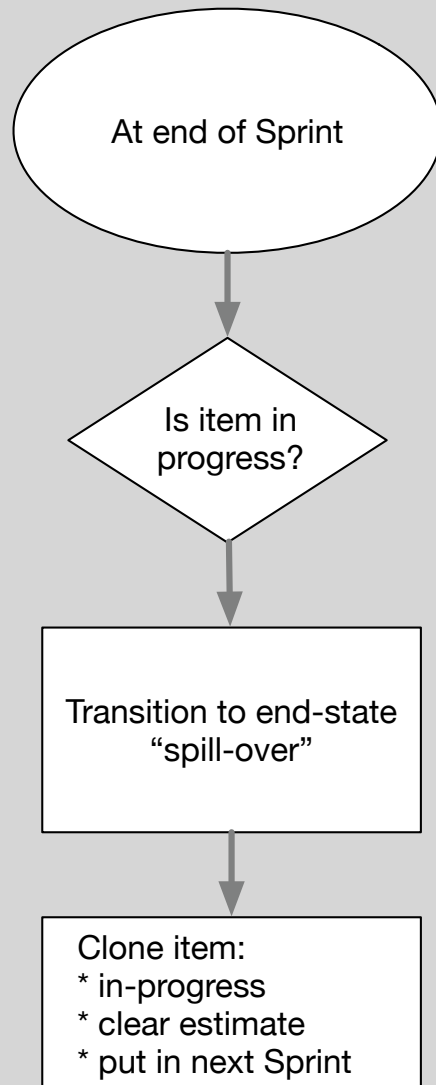
	Items Selected	Items Started During Sprint	Items Done	Spill-over
Example 1	10	5	4	1 (not 6!)
Example 2	1000	3	3	0
Example 3	10	10	6	4



NOT ABOUT PREDICTABILITY!

Not related to say-do metric or planning accuracy in any way

Easy to measure



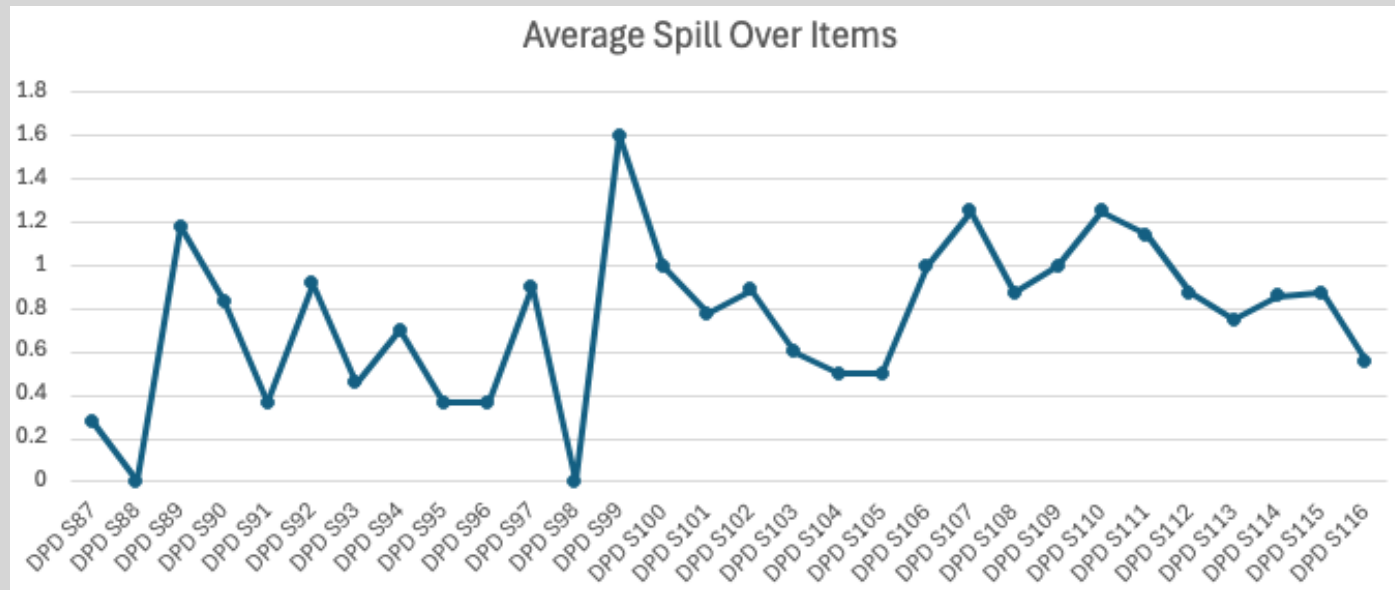
- Spill-over is end-state

- Spillover metric:

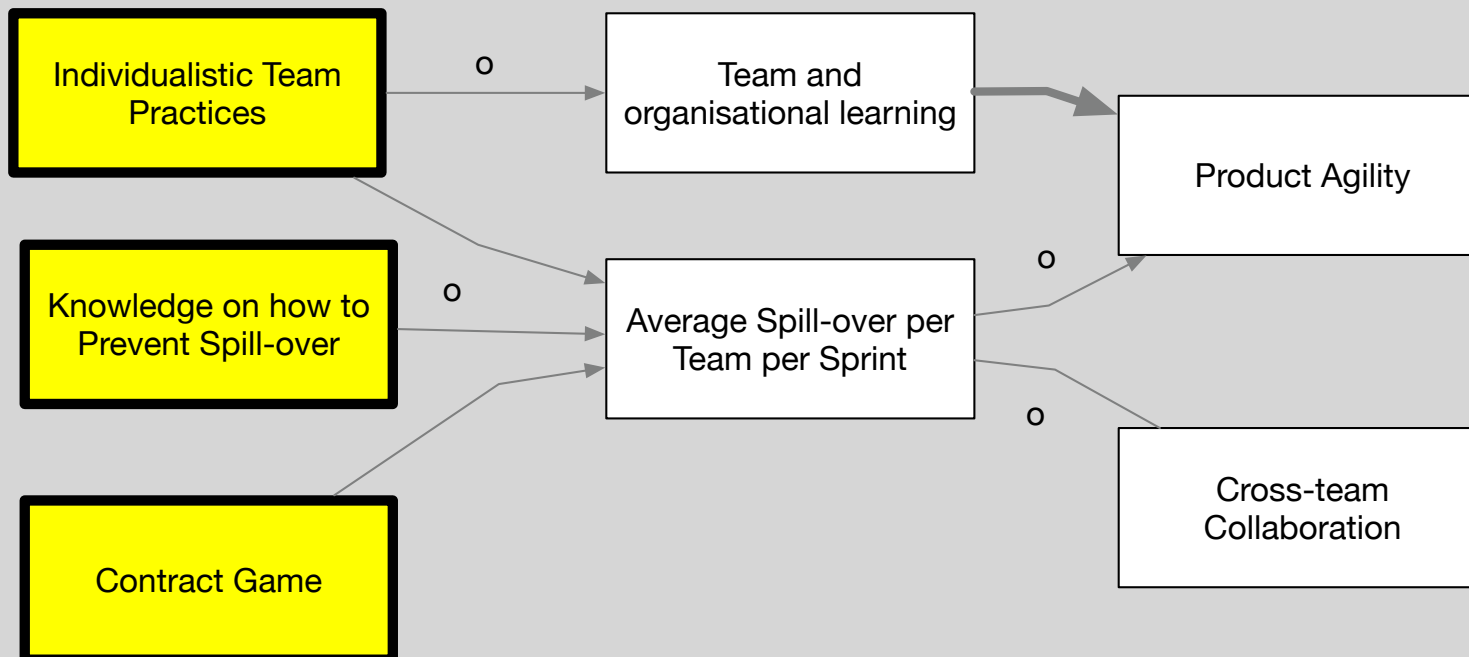
for each item in Sprint x
count spillover state

Measure Spill-over in LeSS

Total spill-over Product /
amount of teams

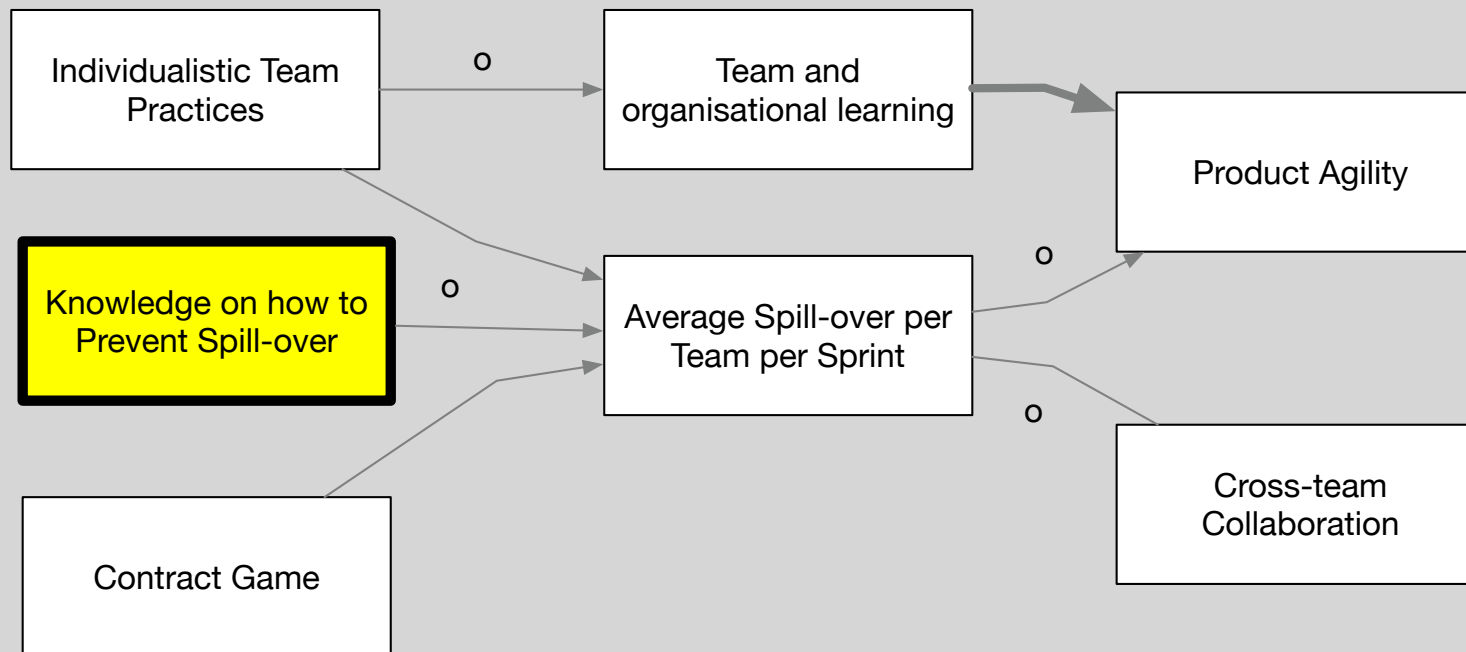


What causes spill-over?



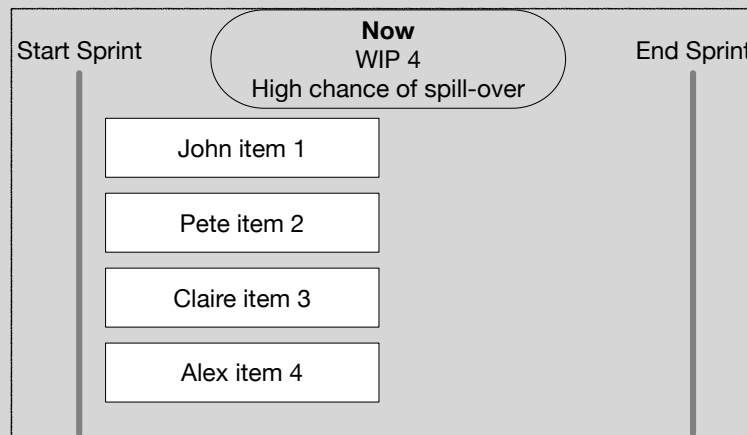
Spill-over can be prevented!

(most of the time)

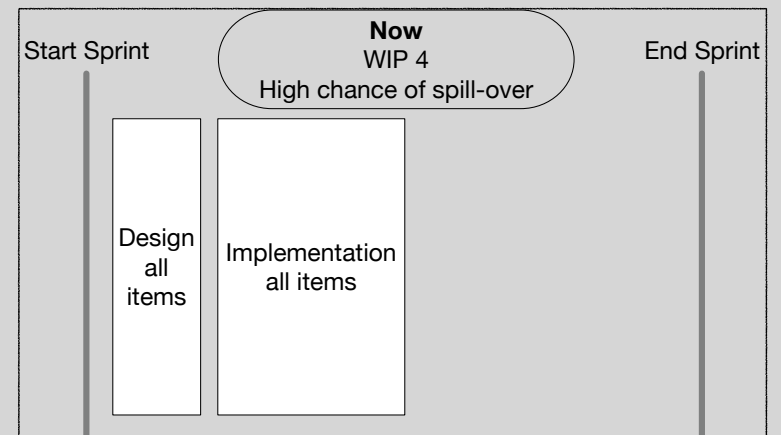


Prevention tactic 1: Reduce WIP

High WIP Examples

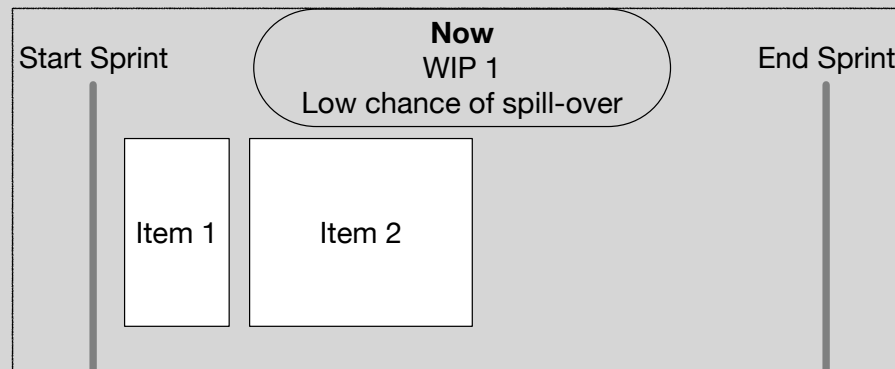


Example 1: Each member has their own items



Example 2: WaterScrum

Low WIP Examples



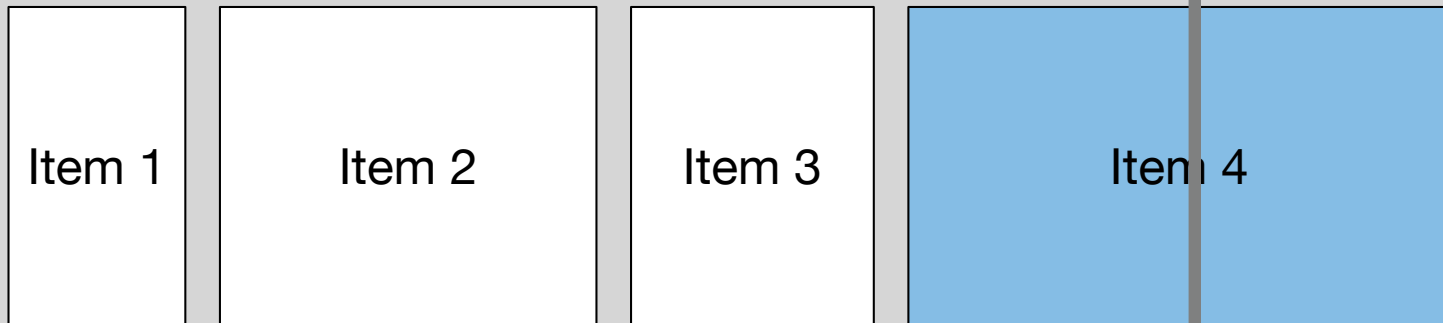
Example 3: One item at the time



Prevention tactic 2: Split or Swap

Start Sprint

End Sprint



Now

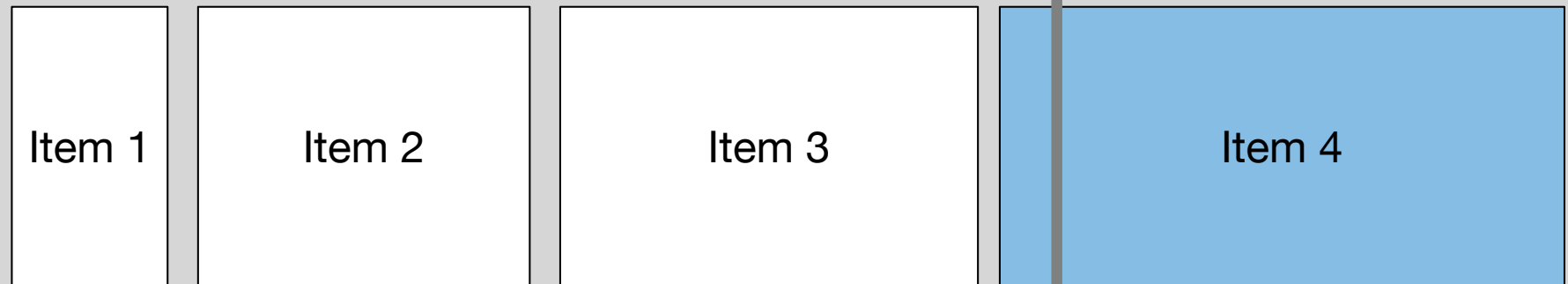
Do not start item 4!
Either split or swap
in-Sprint



Prevention tactic 3: Don't start last day!

Start Sprint

End Sprint



Now

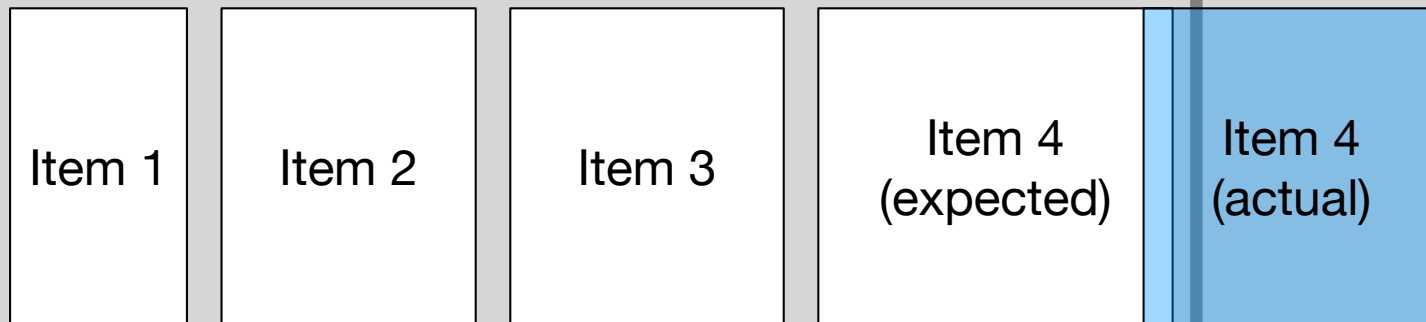
Do not start item 4!
Write a test, learn
something, refactor,
automate something



Tactic 4: Oh Fuck!

Start Sprint

End Sprint

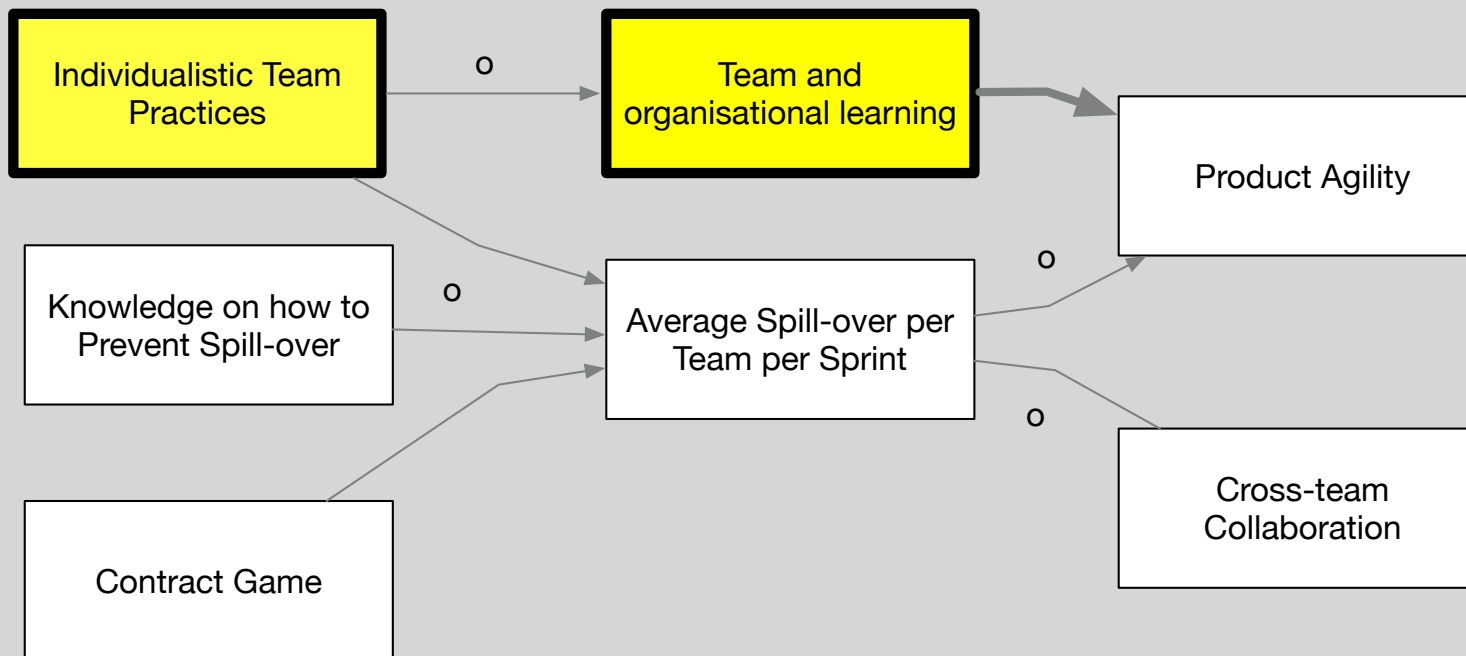


Now

Oh fuck. On the last day
we discovered this!

Don't worry. It happens!
Max spill-over: 1

Team dynamics!





Avoid: Individualistic internal team practices

- Each member has their own items
- Sequential process with functional specialization
- Each member sticks to their single skill
- Feature hogging, also known as “What Sprints?”
- Blocked by dependencies

Each practice leads to **high WIP** and **low collaboration** and **low knowledge sharing**

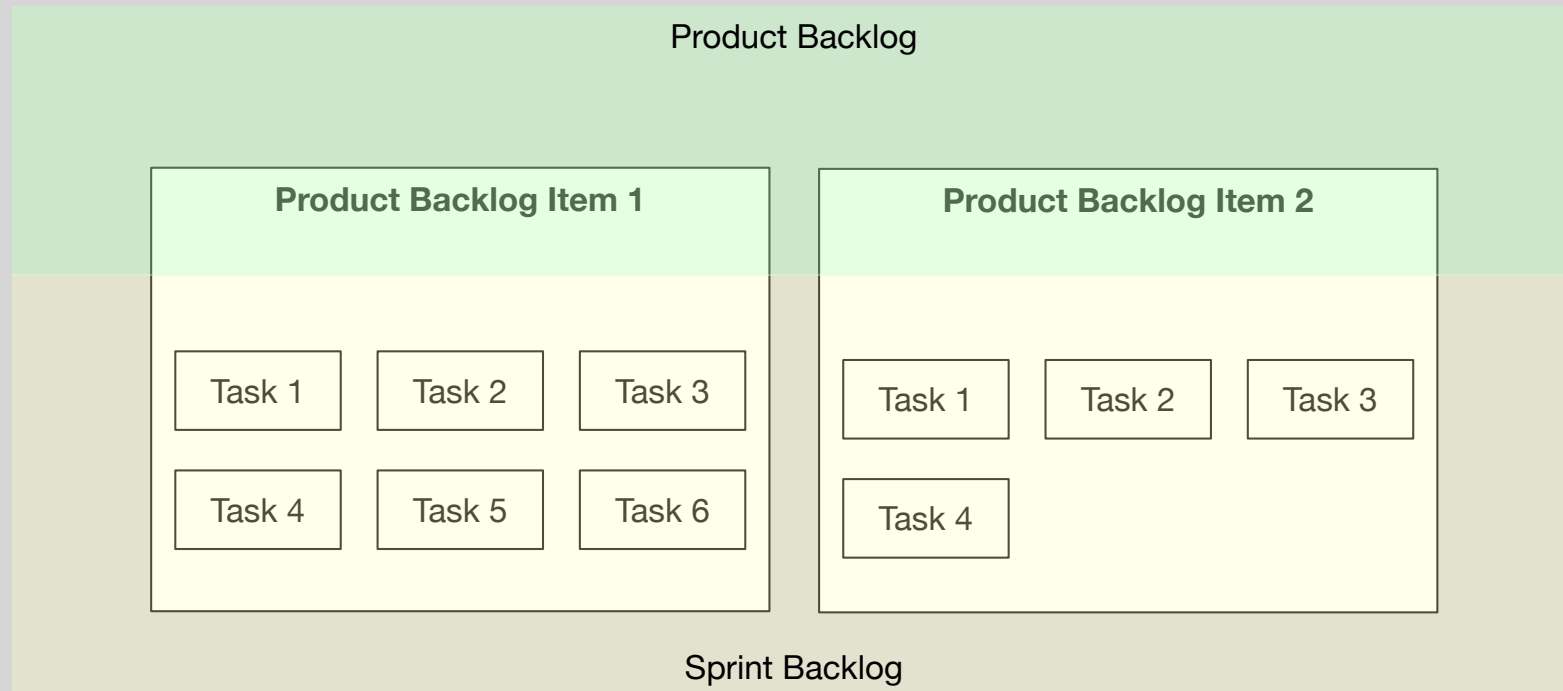


Try: Collaborative internal team practices

- Design in Sprint Planning 2 and fine-grained tasks
- Multiple daily
- Multiple pairs
- Mob programming
- Swarming
- Sacrifice a team member

Each practice leads to **low WIP** and **high collaboration** and **high knowledge sharing**

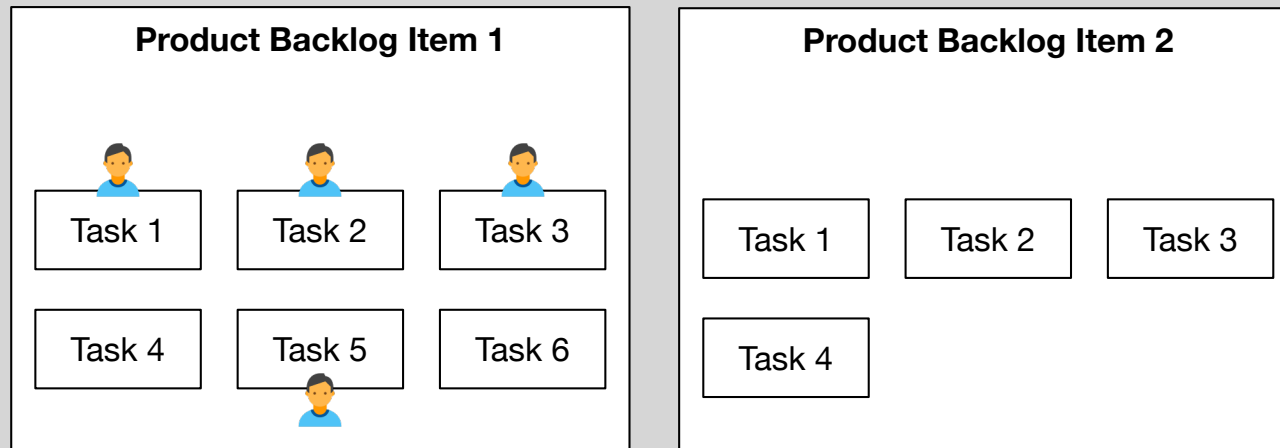
What WIP?



We care most about WIP on Product Backlog level as that affects organisational agility



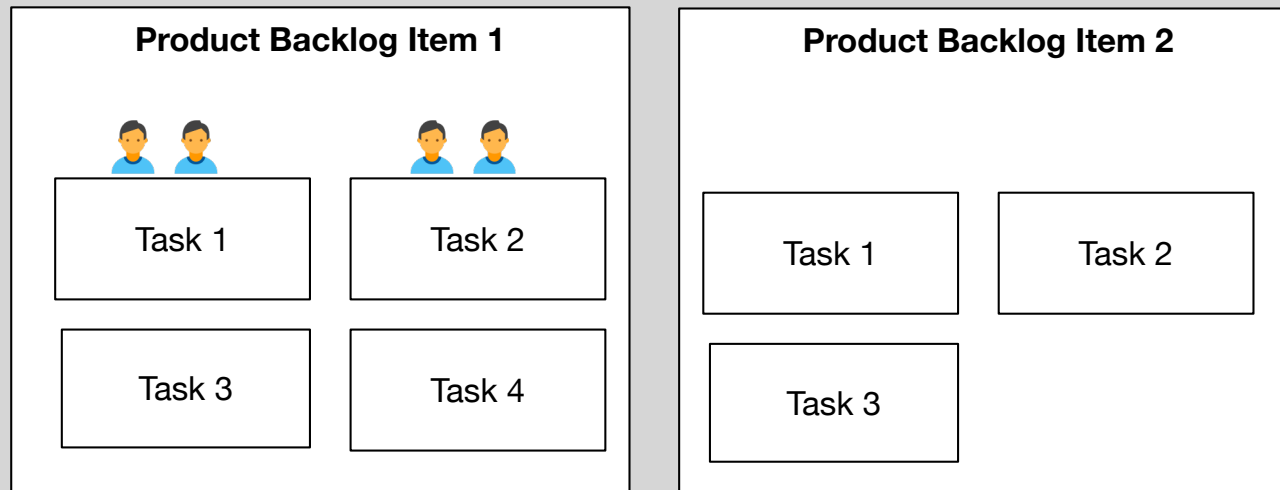
Design in Sprint Planning 2 and fine-grained tasks



- Design in Sprint Planning 2 gives everyone the same shared context
- Fine grain tasks give the ability to split work
- **Multiple daily** give additional sync points



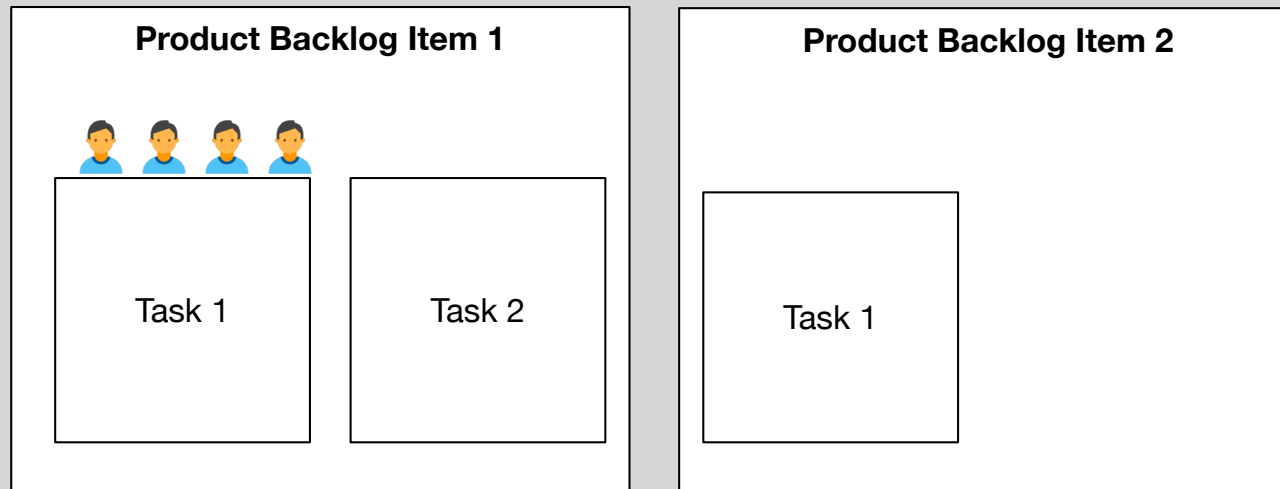
Multiple Pairs



- Reduces synchronisation needs
- Tasks can often be larger



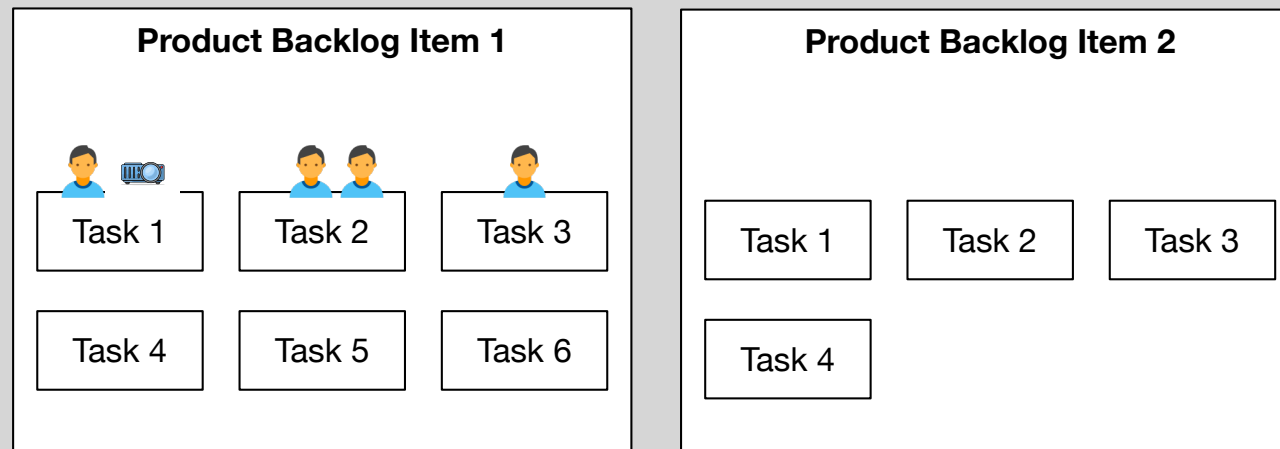
Mobbing (ensemble)



- Item and task WIP 1 (!)
- Often not much Sprint Planning 2 needed

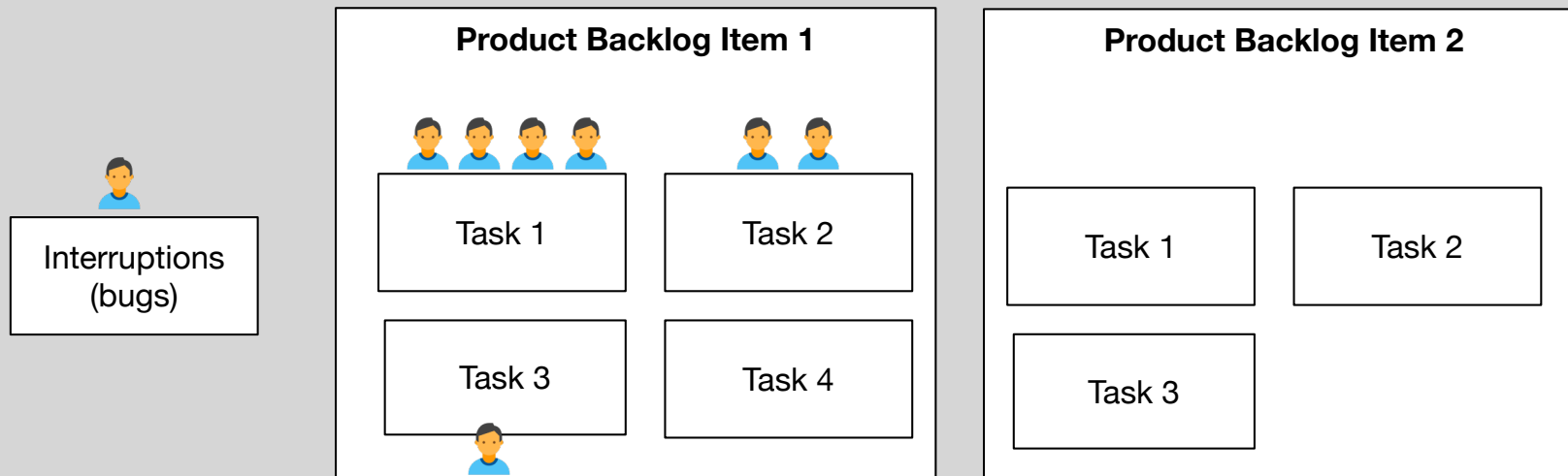


Swarming



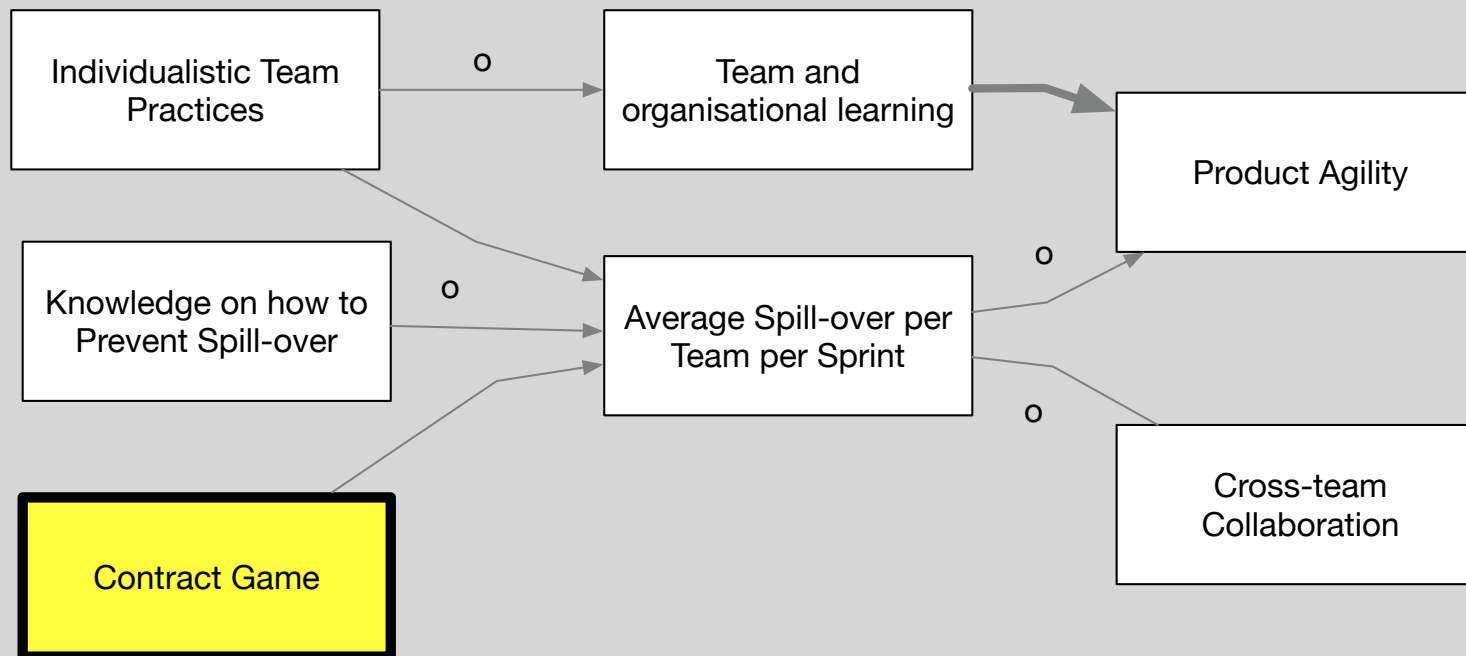
- Whole team in same space
- One central point for synchronisation and overview
- Individuals and pair on separate tasks

Combining (!)



- Teams often combine different practices
- **Sacrifice a team member** can prevent interruptions to the whole team

Contract Game





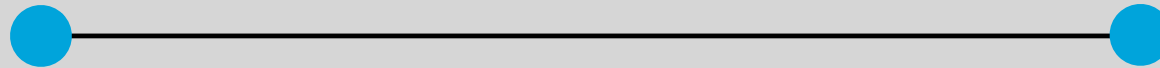
Product
Management



R&D

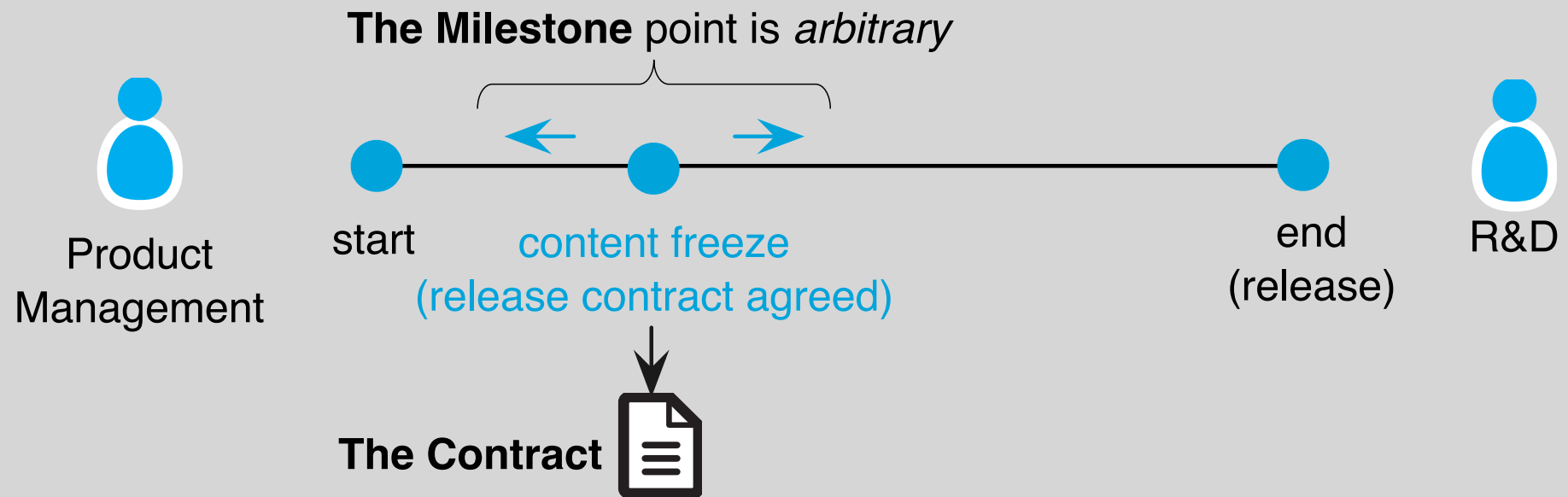


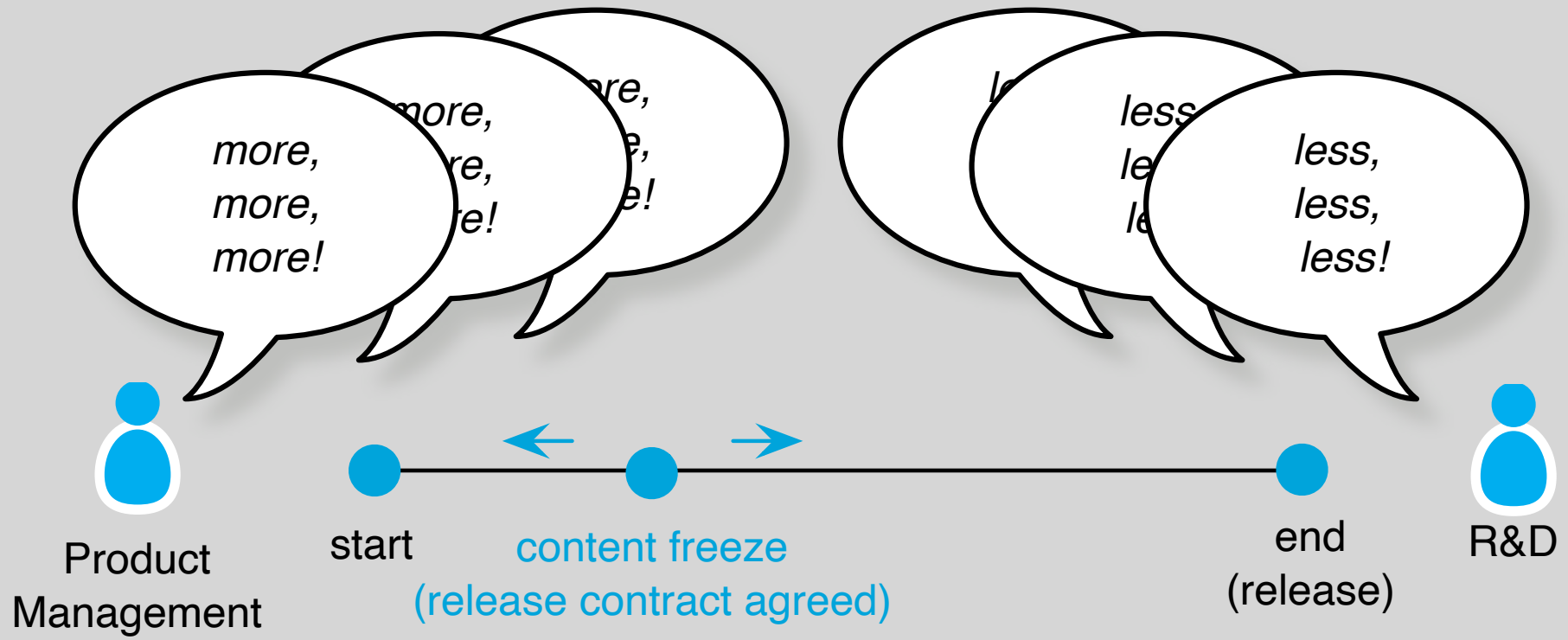
start



end
(release)









Deadlines / Promises

Why Deadlines and Promises?	
Hard external constraints (e.g. regulator or end-of-life)	
Synchronisation (e.g. marketing campaign or process change)	
Promised to customer (e.g. part of a deal that was sold)	
Motivate (usually random deadline)	

Deadlines / Promises

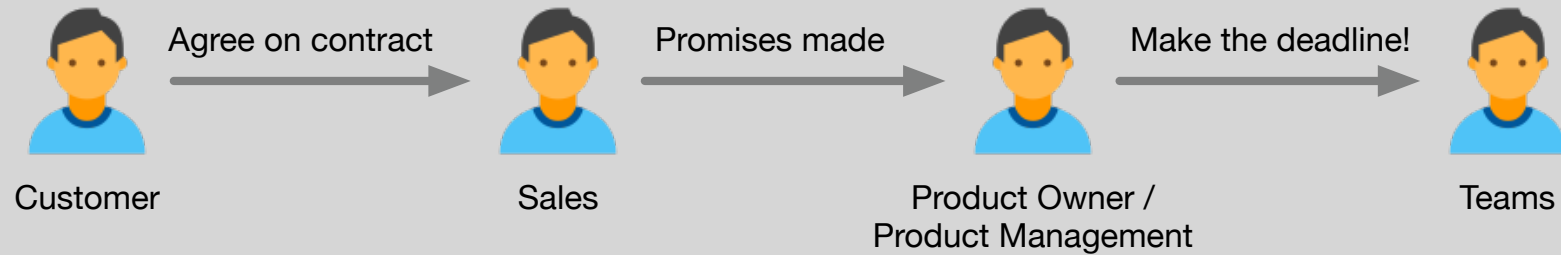
Why Deadlines and Promises?	Why deadlines change?
Hard external constraints (e.g. regulator or end-of-life)	Different scope for meeting constraint or temporary workarounds
Synchronisation (e.g. marketing campaign or process change)	Delay in dependency
Promised to customer (e.g. part of a deal that was sold)	New opportunity with higher impact than previous promises
Motivate (usually random deadline)	Surprises discovered in development

No plan-driven predictability

Predictability through following plan in dynamic environment leads to rigidity and unrealistic product- and people-destroying commitments

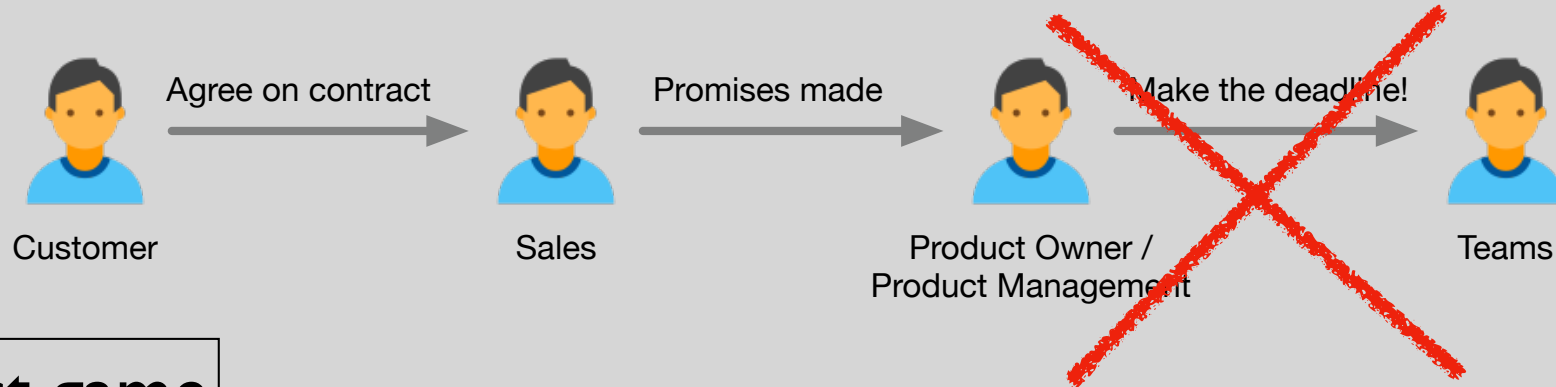
Predictability through continuous reprioritisation leads to not being predicable in plan but in outcome

Dealing with Deadlines



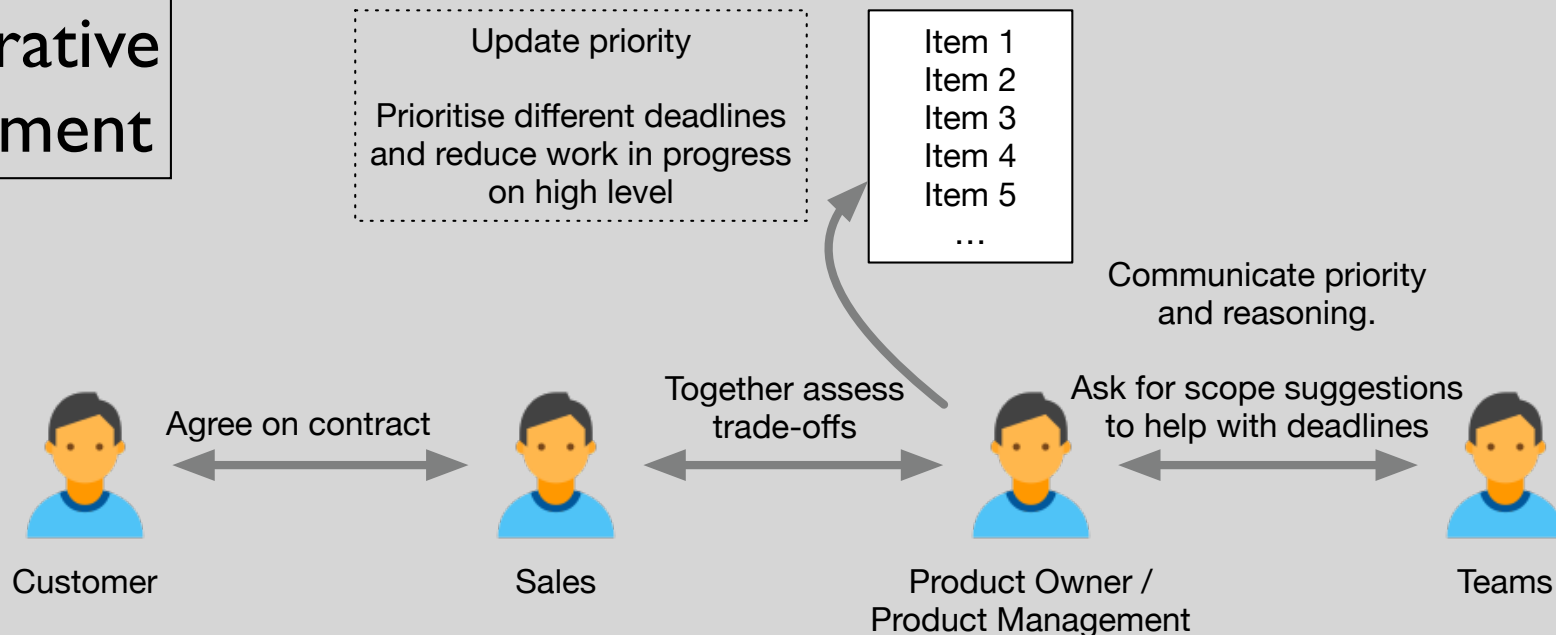
Contract game

Dealing with Deadlines



Contract game

Collaborative development





Scheduling

Team A	Item for #1	Item for #1	Item for #1	Item for #1	Item for #5	Item for #5	Item for #5	Item for #5	Item for #5
Team B	Item for #2	Item for #2	Item for #2	Item for #2	Item for #2	Item for #2	Item for #2	Item for #2	Item for #6
Team C	Item for #3	Item for #3	Item for #3	Item for #3	Item for #3	Item for #3	Item for #3	Item for #3	Item for #3
Team D	Item for #4	Item for #4	Item for #4	Item for #4	Item for #4	Item for #4	Item for #4	Item for 4	Item for #4

Contract game

Deadline 1

Deadline 2



Scheduling

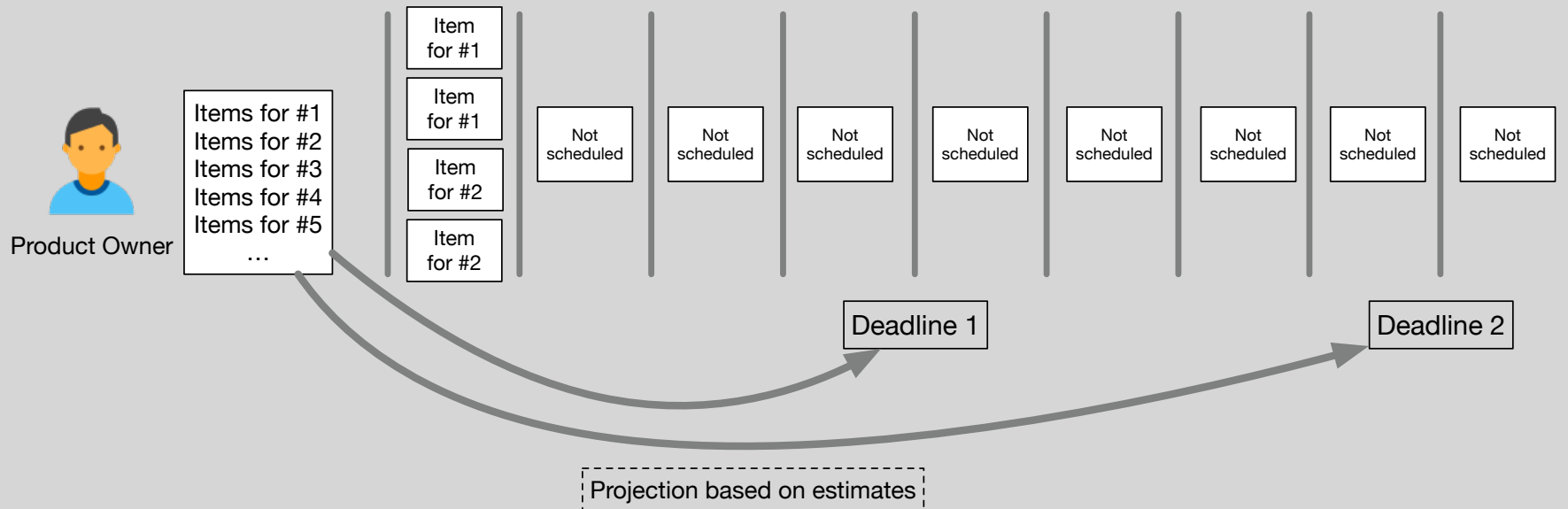
Team A	Item for #1	~~Item for #1~~	Item for #1	~~Item for #1~~	Item for #5	~~Item for #5~~	~~Item for #5~~	Item for #5	~~Item for #5~~
Team B	Item for #2	Item for #2	~~Item for #2~~	Item for #2	~~Item for #2~~	~~Item for #2~~	Item for #2	~~Item for #2~~	Item for #6
Team C	Item for #3	Item for #3	~~Item for #3~~	Item for #3	~~Item for #3~~	~~Item for #3~~	Item for #3	~~Item for #3~~	Item for #3
Team D	Item for #4	~~Item for #4~~	Item for #4	~~Item for #4~~	~~Item for #4~~	Item for #4	~~Item for #4~~	Item for #4	~~Item for #4~~

Contract game

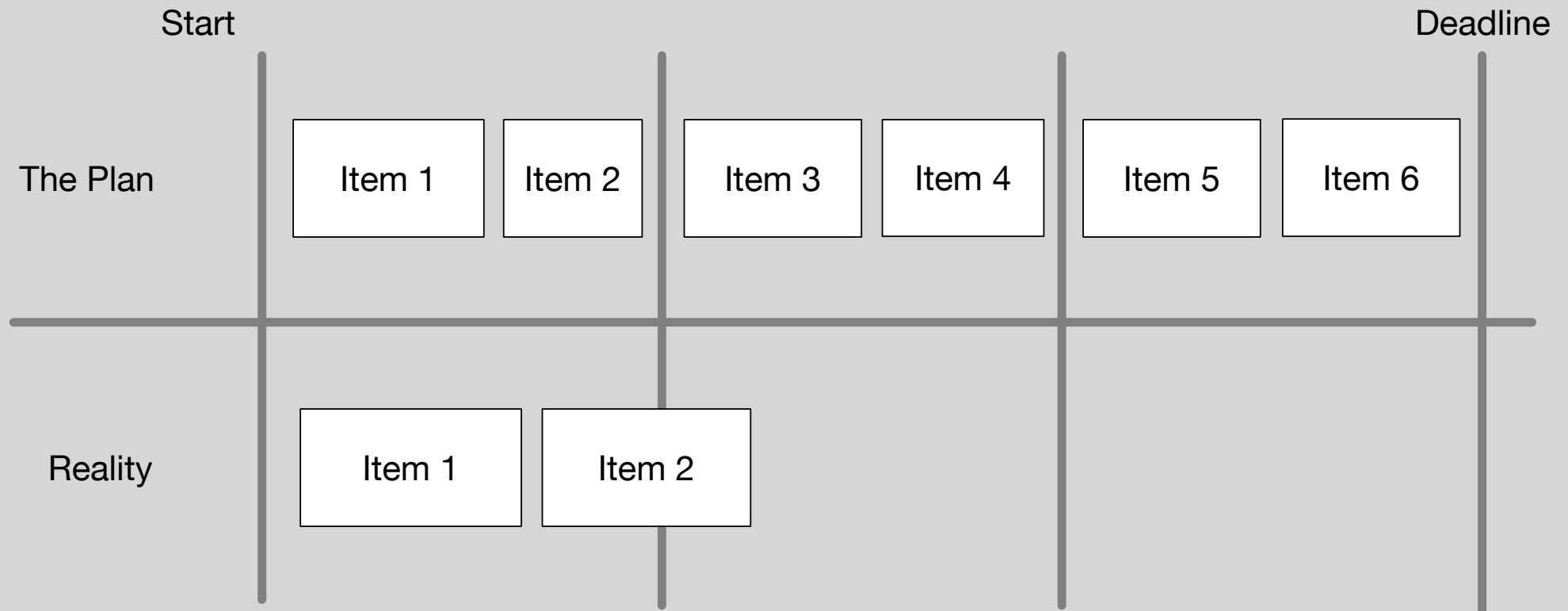
Deadline 1

Deadline 2

Collaborative development



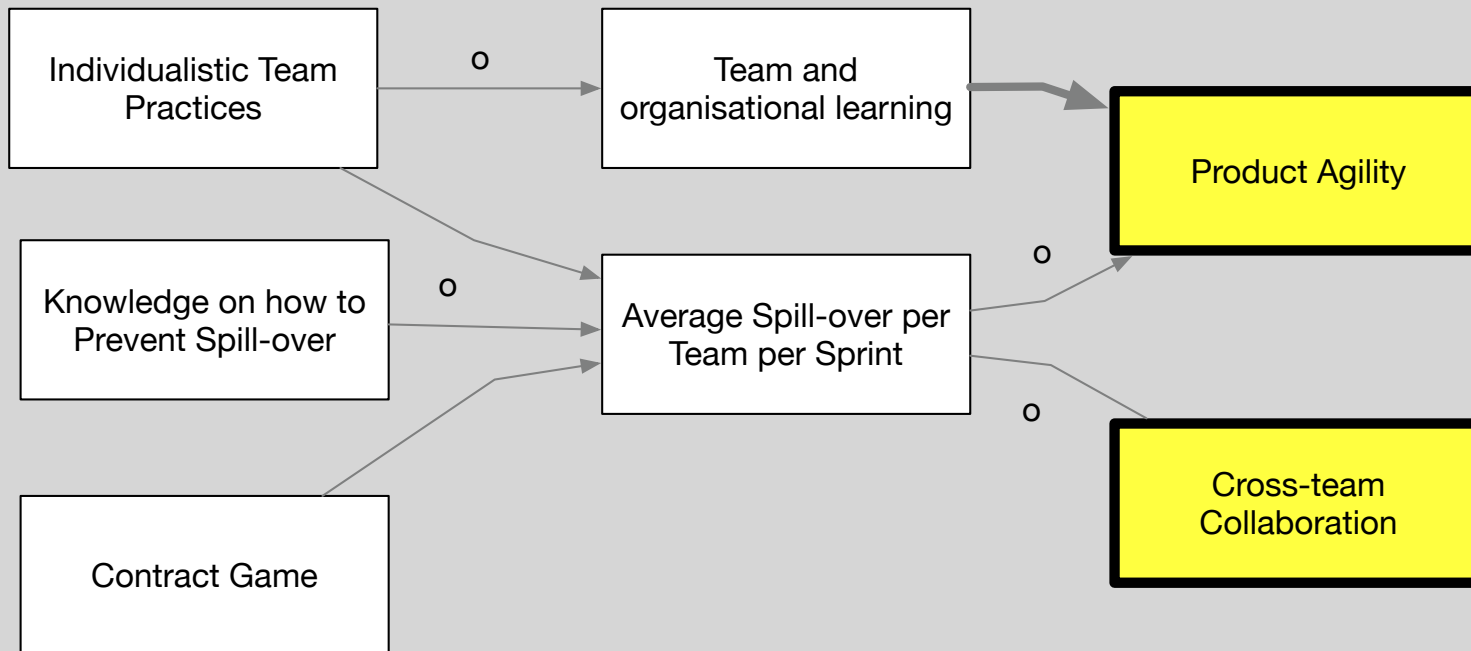
Contract Game - Spillover



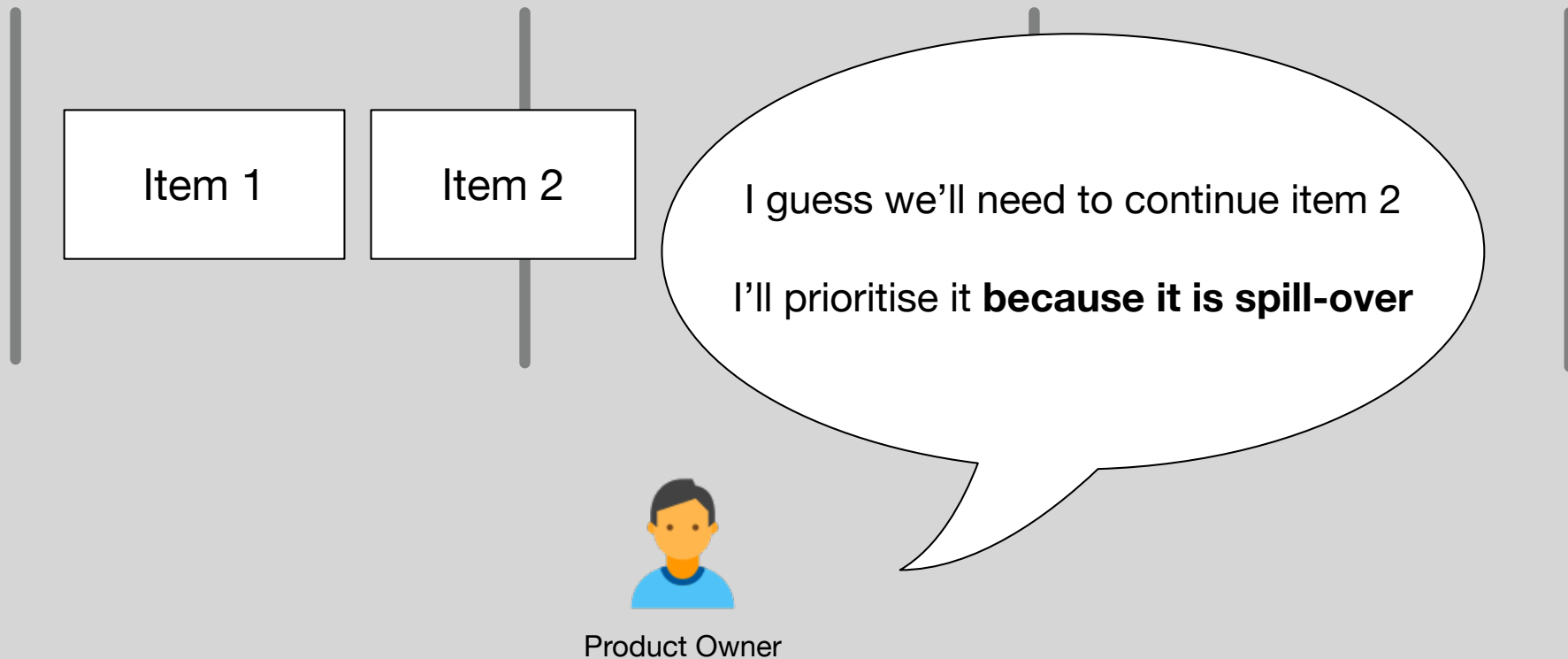
Why bother!
Spill-over, who cares!

These iterations are fake

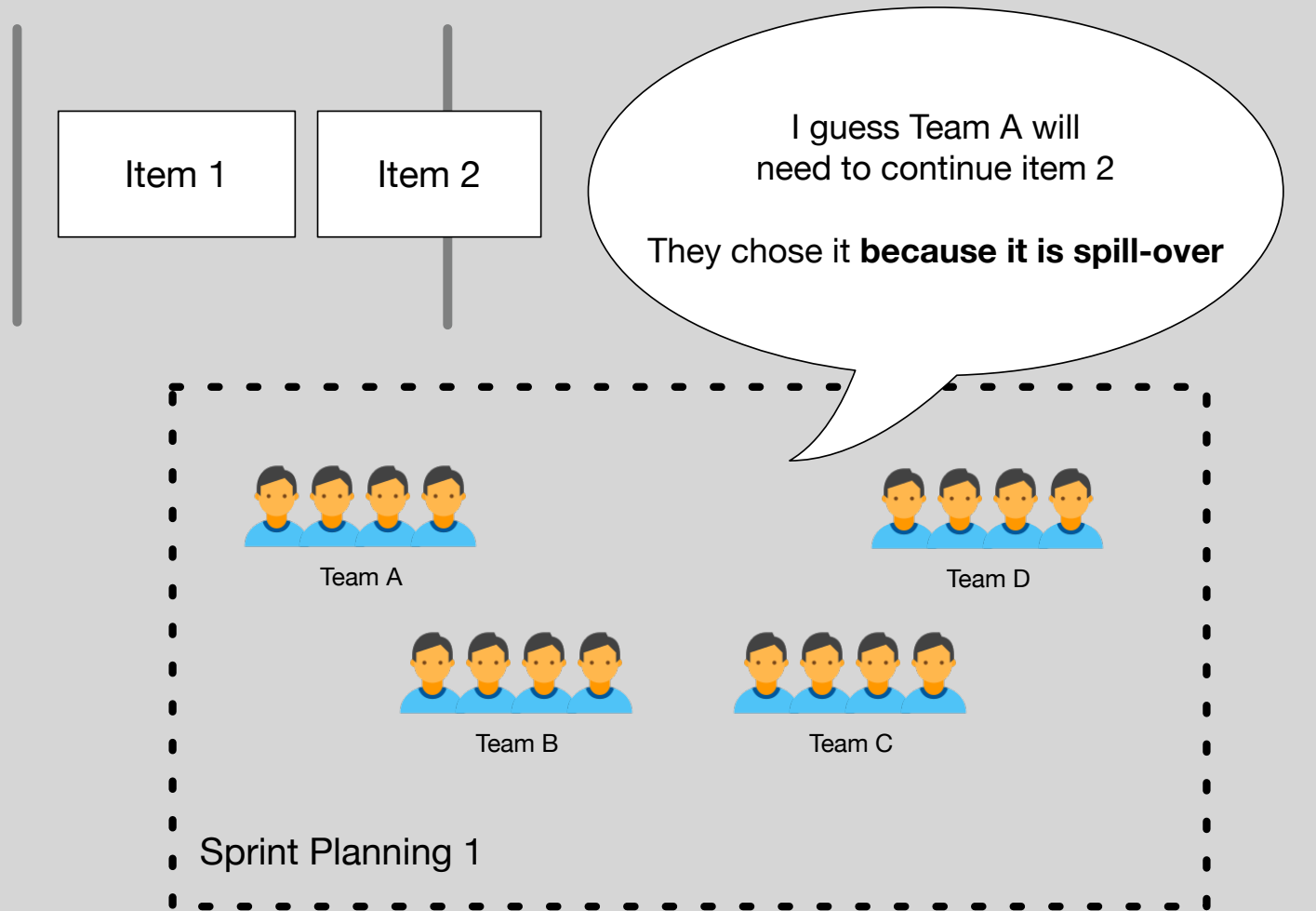
What causes spill-over



Product Agility



Cross-team Collaboration





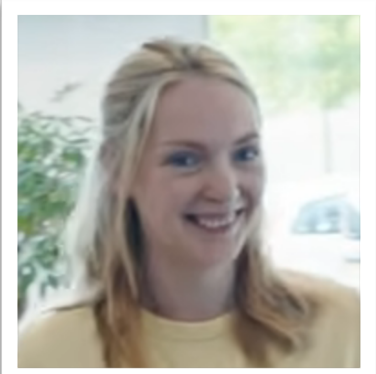
Summary (1)

- Measuring spillover does not only measure spillover but also:
 - How well teams are collaborating
 - Whether there are contract game dynamics ongoing
 - Whether teams understand their choices and responsibilities during the Sprint



Summary! (2)

- Therefore, managing to lower spillover usually leads to:
 - Increased team collaboration
 - Increased PO flexibility
 - Increased cross-team collaboration



Why are you
always so obsessed
with spill-over?

It is an **easy to
measure** metric and focus on
it leads to **increased in- and
cross-team collaboration** and
**healthy Product
Ownership**

