Collaboration and Contracting with Partners in Large Agile Development

Influence on Innovation and Product Development

michael.mai@valtech.com

Michael Mai

- LeSS Coach
- Coach for Technical Excellence
- Working with 1 to >40 Teams
- michael.mai@valtech.com
- http://www.agilesoftwaredesign.de/

Partner?

- Party that cannot or will not fully integrate in your development process / working model
  - Possibly with additional contracting in place

- Large groups
  - Other departments

- Contractors

- Large partners
  - Big size partners those investing an roughly equal share of effort, equal share in economics risk, entrepreneural decisions, own economic benefits, …

01

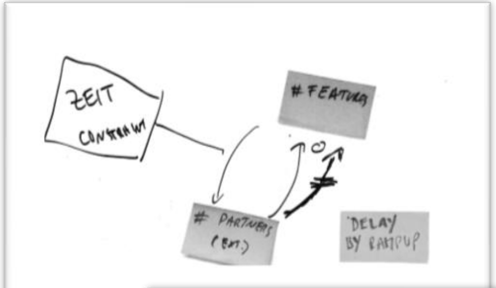
section

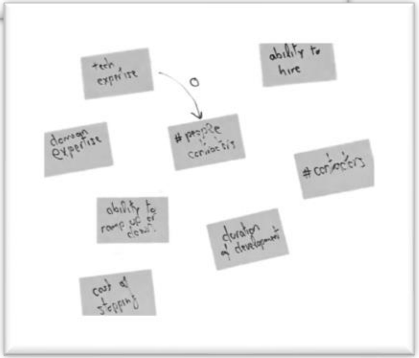
Business

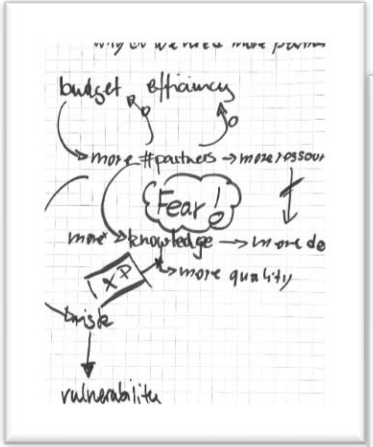
Why adding partners to the effort?

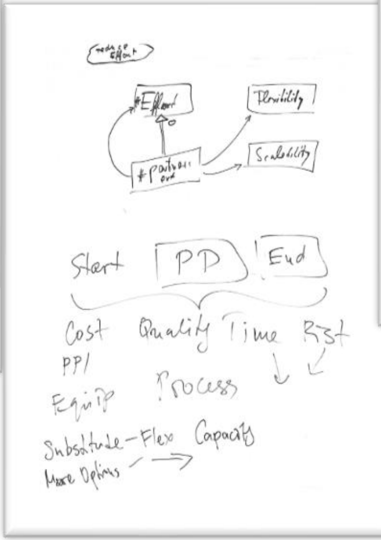
What is the optimizing goal and constrains

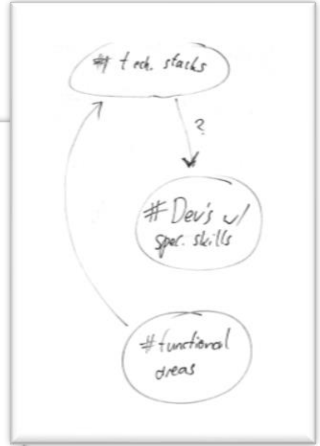
- Models by the participants of the LeSS Meetup 2019-02-21











valtech_

In the benefit for the company

- There is no static world
  - Continued ability to adapt to market change

- Uphold high brand image
  - Fast reaction to discovered “rough-edges” in user workflow
  - Fast reaction to discovered “bugs” or undesired behaviors

- Keeping options open
  - E.g. ability to phase-out the product in favor to a new product

02

section

Legal consideration

considering German Labor Law and Company Law

Neither Valtech nor I do provide legal counsel

> *If you say, you contract an independent company. The contractor need to stay independent.*
>
> *German law for beginners*

valtech_

Independent

- I’m independent, if I decide for myself
  - When to work
  - Where to work
  - How to work
- I’m independent, if there is no
  - … strict and legal binding hierarchy restricting my work
- I’m
  - Free make own entrepreneur decisions
  - Take an entrepreneur risk

contributing?

Independent but still contributing

- Okay is
  - Instruction based on the product
  - Instruction based on quality
  - Instruction based on tooling and integration so it adding to the product development effort
  - Collaboration process as interface to the employer

- Difficult
  - Dictating the only one unchangeable processes to follow

Entangled work

Instructions, hierarchies, work output, …

- How to differentiate
  - self-defined close working
  - vs. instructed work
- How to
  - clearly differentiate your work output as an result of an independent effort
- How to
  - be more than the sum

Why the fuss?

- Audit on actual processes not on solely contracts
  - Walk the talk
- Employer may be verdicted with an economic crime
  - Possible exclusion from tenders
  - Since 2017
- No a-priori pardon
  - You need to state the concrete collaboration details prior to begin of work
  - Since 2017

Different aspects

… not explored

- Staff leasing (German: “Arbeitnehmerüberlassung”)
- Bogus self-employment (German: “Scheinselbständigkeit”)
- Depreciation (German: “Steuerliche Abschreibung”)
- Service contract vs. contract for work

03

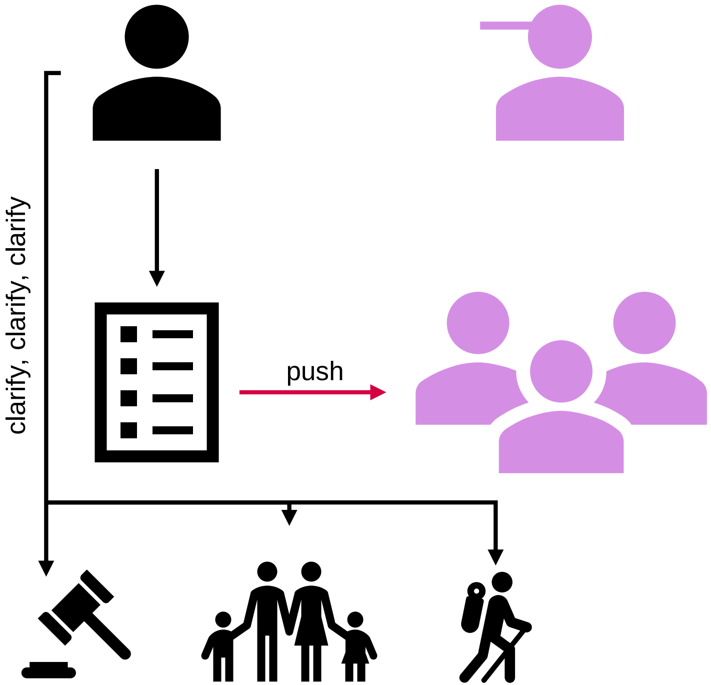
section

Starting small and simple:

One Team Product Development

Setup

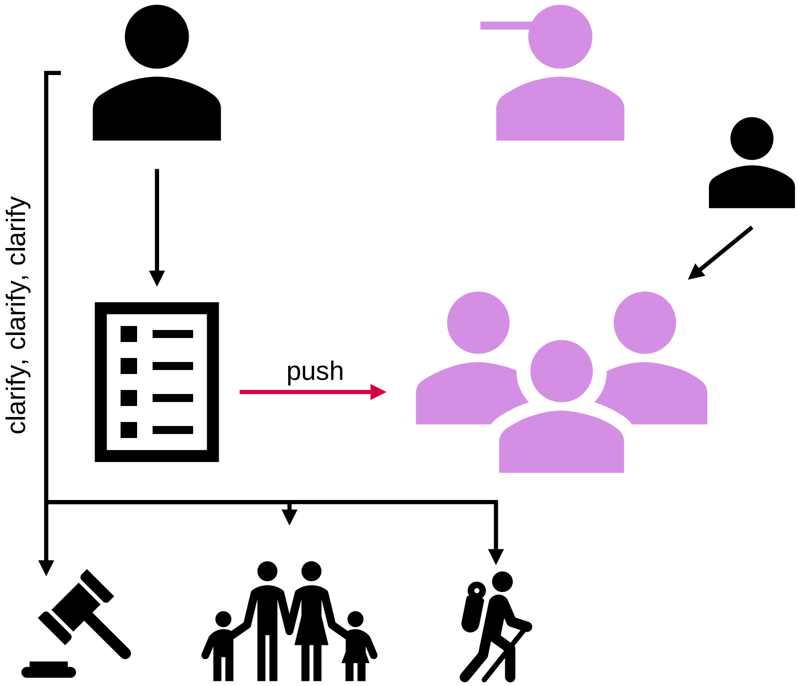
highly critical



- PO provide massive information on details
  - □ Team is deprivated from customer clarification
- PO direct development
  - □ via “Why” and characteristic of the product
  - □ via the Product Backlog
  - □ via massive details in item, comments, emails, meeting
- Teams compiles a Sprint Backlog by themselves
- Team is directed directly by PO via assignment of items in the backlog

Setup

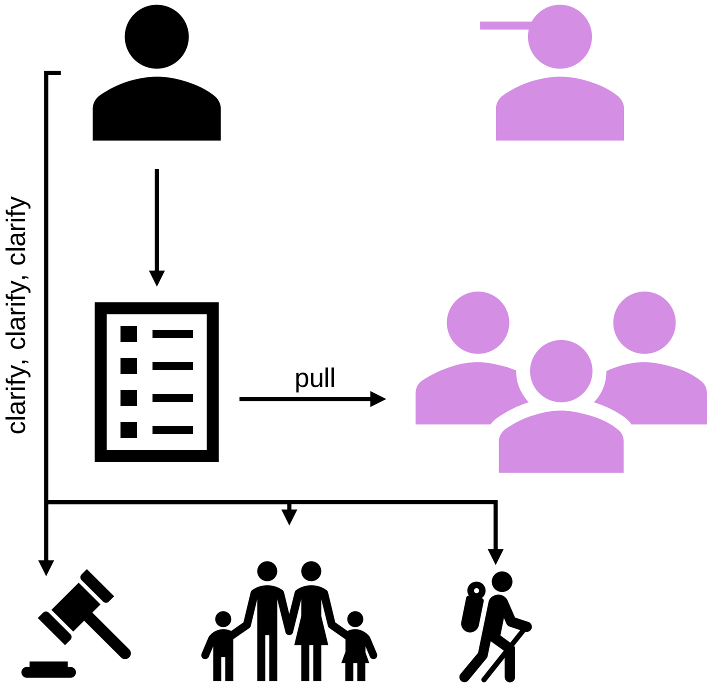
highly critical



- PO provide massive information on details
  - □ Team is deprivated from customer clarification
- Team receives direction not via backlog
- PO direct development
  - □ via “Why” and characteristic of the product
  - □ via the Product Backlog
  - □ via massive details in item, comments, emails, meeting
- Teams compiles a Sprint Backlog by themselves
- Team is directed directly by PO via assignment of items in the backlog

Setup

critical

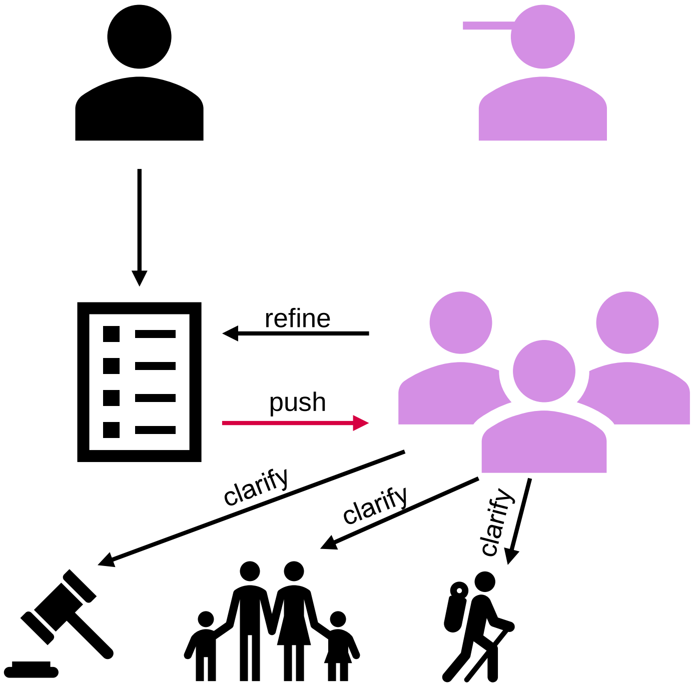


- PO provide massive information on details
  - Team is deprivated from customer clarification
- PO direct development
  - via “Why” and characteristic of the product
  - via the Product Backlog
  - via massive details in item, comments, emails, meeting
- Teams compiles a Sprint Backlog by themselves
- Team pulls items based on ordering in product backlog by themselves

valtech_

Setup

critical

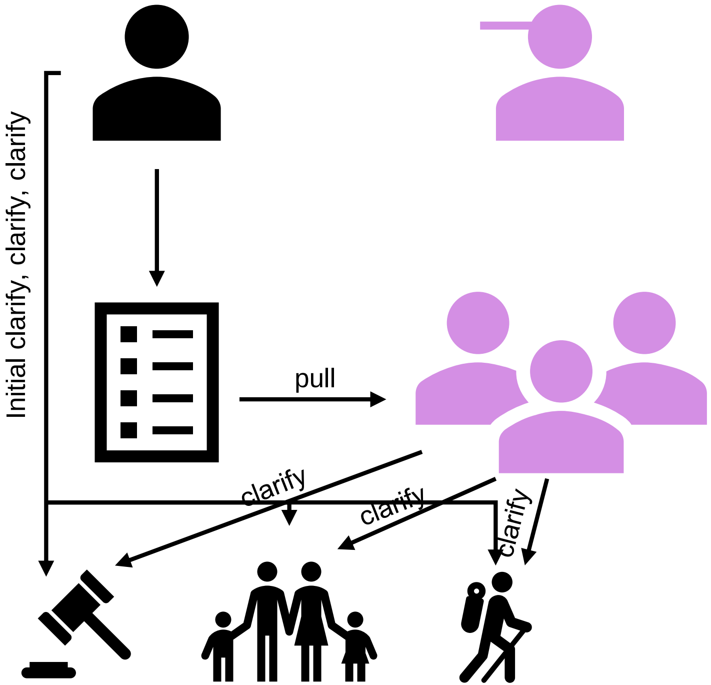


- Team refines items themselves
- Team clarify details themselves
- PO direct development
  - via “Why” and characteristic of the product
  - via the Product Backlog
- Teams compiles a Sprint Backlog by themselves
- Team is directed directly by PO via assignment of items in the backlog

Setup

critical

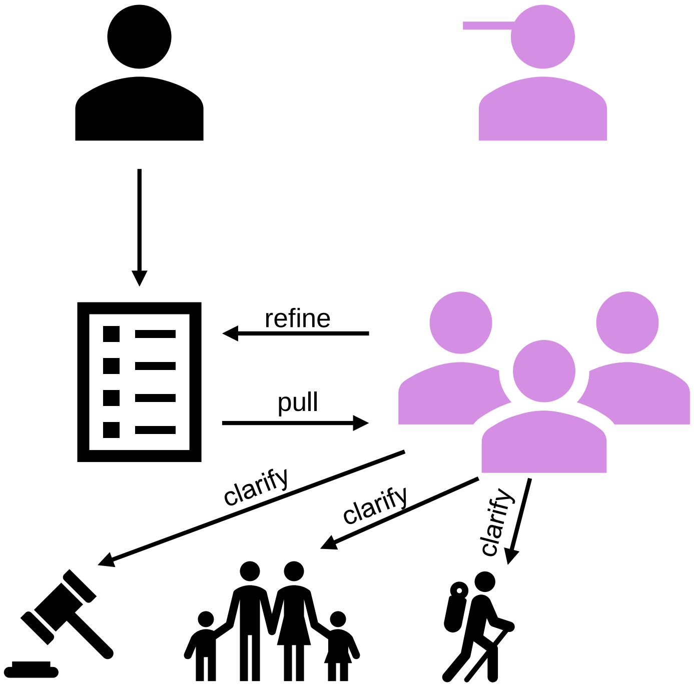
- This is up the creek



valtech_

Setup

okay



- Team refines items themselves
- Team clarify details themselves
- PO direct development
  - via “Why” and characteristic of the product
  - via the Product Backlog
- Teams compiles a Sprint Backlog by themselves
- Team pulls items based on ordering in product backlog by themselves

Product Backlog vs. To-Do lists

Legally import differentiation

- Product Backlog
  - “The Product Backlog is an ordered list of everything that is known to be needed in the product” (https://www.scrumguides.org/)
  - “[…] Product Backlog that defines all of the work to be done on the product. They [Teams] do not each have their own Product Backlog. Product Backlog Items are not pre-assigned to the teams.” (https://less.works/)
  - → product focus
- (dynamic) To-Do list
  - No necessary product focus
  - → no product focus, therefore risk of focus on “how and what” and not of “why”

valtech_

Instructions

Scope and Context matters

- Instructions based on product leave from for independent decision that still contribute to the product development effort
  - Directions on product level via product backlog → okay

- Some translations
  - Clarification → “Auftragsklärung”

Instructions

Clarification	Assignment	
By team	By team	Okay
By team	By PO	Critical
By PO	By team	Critical
By PO	By PO	Highly critical
By *	By Scrum Master	Critical

Prioritization

Prioritization	
By team	Okay
By PO	Okay
By Scrum Master	What?

How to contract this?

Assuming contract for work

- Refinement contract
  - Deliverable
    - Refined product backlog items

- Sprint contract
  - Deliverable
    - Necessary work as defined by Definition of Done
    - Outcome from Retrospective as a prove to improve the own processes

04

section

Different models

Evaluation schema

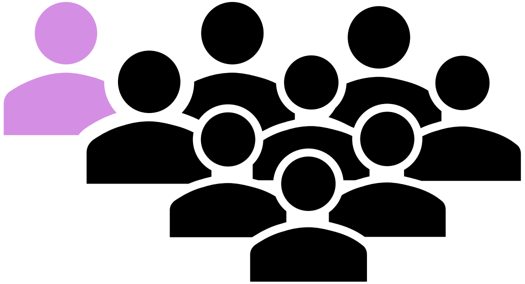
- Company retains product live time
  - Innovatability
  - Maintainability
  - Freedom of direction
  - Freedom of commercial use
- “True” commitment by value worker

What is a team?

- More than one person, less than a crowd
  - Scrum: 3-9 people
- One common goal
  - Scrum: Sprint Goal
- Working collaboratively toward this goal
  - How to arrange this legally?
  - Partner does need to work “independent”/self-managing and may only share work results and information

1 partner in 1 team

1 partner in 1 team

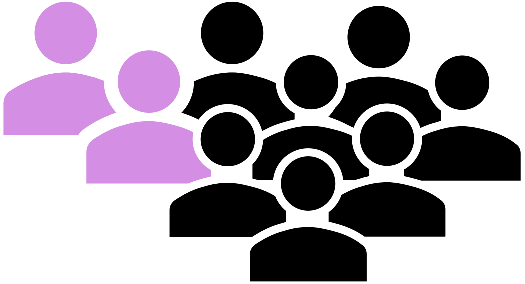


Watch critically for
DOs and DON’T DOs

- Refinement
  - Team refine
  - Partner refine in parallel
  - Join and exchange refinement result
- Planning 1
  - Volunteer for item based on ordering in backlog
  - (A)PO approve or decline selection of teams
- Planning 2
  - Independent SP2 and solution planning
  - Join and exchange
  - Decided by non-hierarchical vote ← critical
- Sprint
  - Team and partner work in parallel (not on the same item)
  - Constant exchange of work results by frequent merge and push on origin/master
  - No pairing and no mob working

2... partner in 1 team

2… partner in 1 team

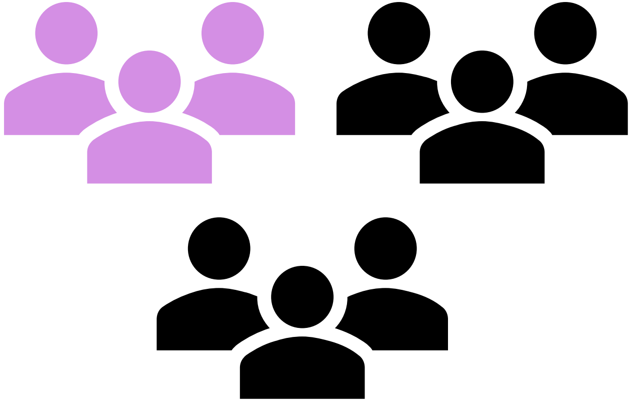


Watch critically for
DOs and DON’T DOs

- Refinement
  - Team refine
  - Partner refine in parallel
  - Join and exchange refinement result
- Planning 1
  - Volunteer for item based on ordering in backlog
  - (A)PO approve or decline selection of teams
- Planning 2
  - Independent SP2 and solution planning
  - Join and exchange
  - No grantee pick for partner
  - Decided by non-hierarchical vote ← critical
- Sprint
  - Team and partner work in parallel (not on the same item)
  - Constant exchange of work results by frequent merge and push on origin/master
  - Two partner may pair work

One partner team

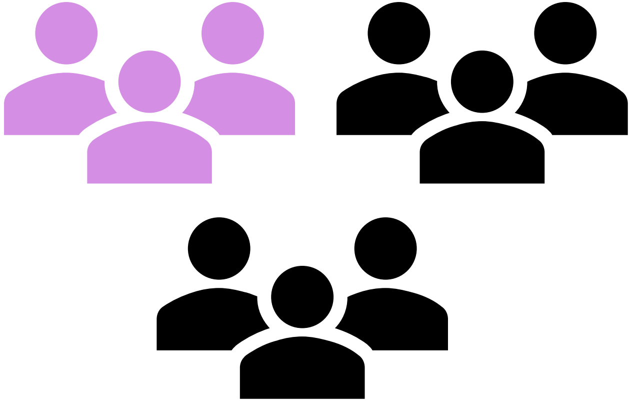
One partner team



Watch critically for
DOs and DON’T DOs

- Refinement
  - All teams refine, no mix of partner and non-partner team during multi-team PBR
  - Refinement also defines the product → you may want to direct the refinement, legally not needed
- (Special) Refinement
  - Employer provides headlines for refinement
    - May be provided by non-partner teams
  - Partner team refine within the predefined headlines
    - specially devised contract, to refine only the headlines
    - “real” refinement

One partner team



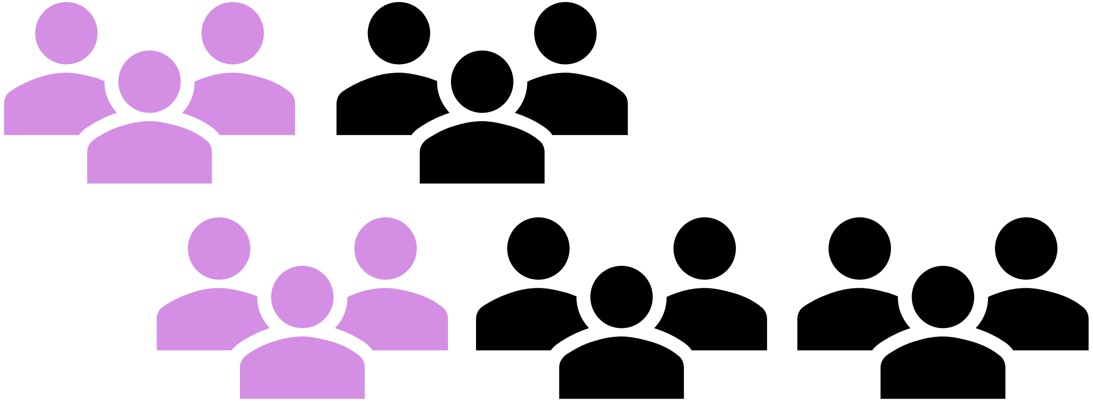
Watch critically for
DOs and DON’T DOs

- Planning 1
  - Volunteer for item based on ordering in backlog
  - (A)PO approve or decline selection of teams
- Planning 2
  - Done within each team individually
- Sprint
  - Done within each team individually
  - Information exchange allowed
  - no collaborative work on same item allowed
  - Constant exchanging work results by frequent merge and push on origin/master

Many partner team

Many partner teams

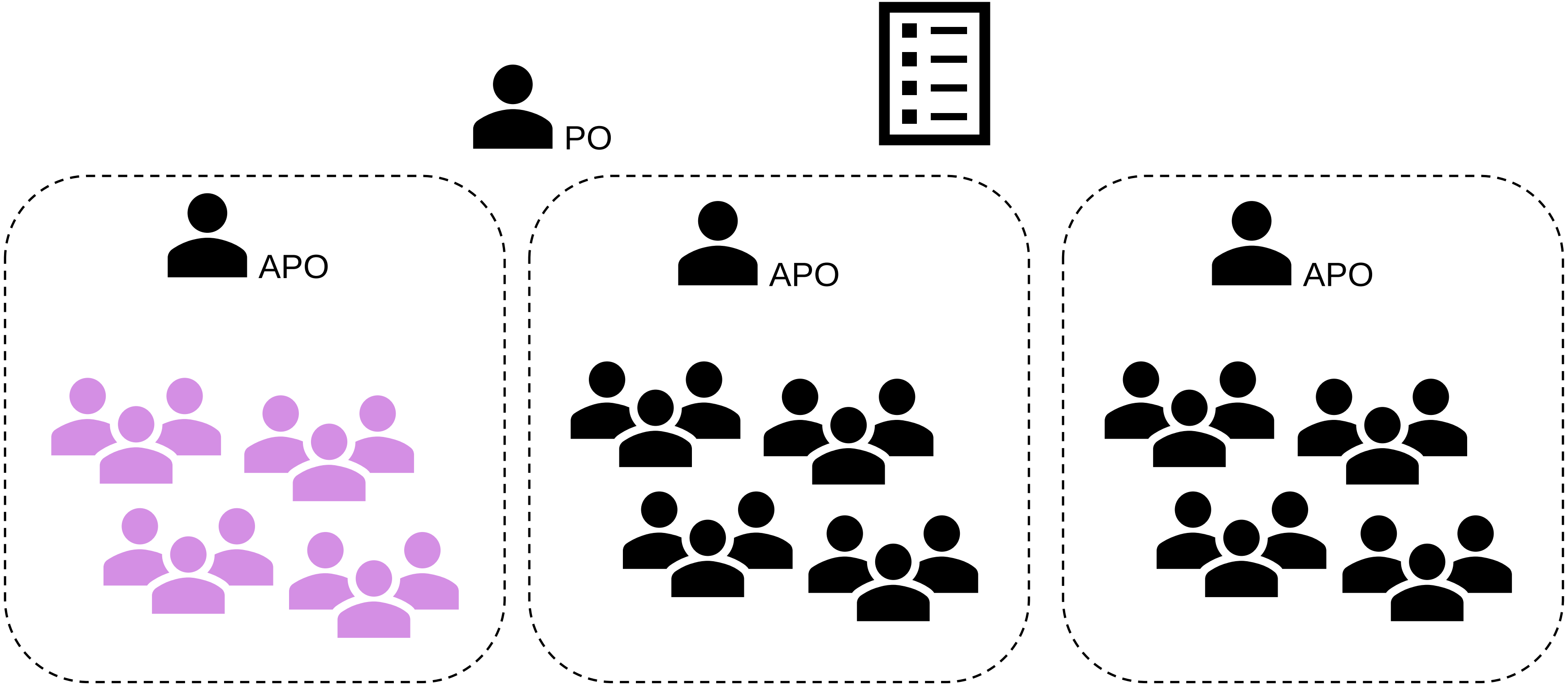
- Almost the same as in “One partner team”



Watch critically for
DOs and DON’T DOs

One partner requirement area

One partner requirement area



One partner requirement area

- Most refinements are conducted within each requirement area
- Information exchange between areas
  - Within same partner → okay
  - Within same employer → okay
  - Refinement mix between partners
    - Pure information exchange → okay
    - Collaboratively create information → highly critical

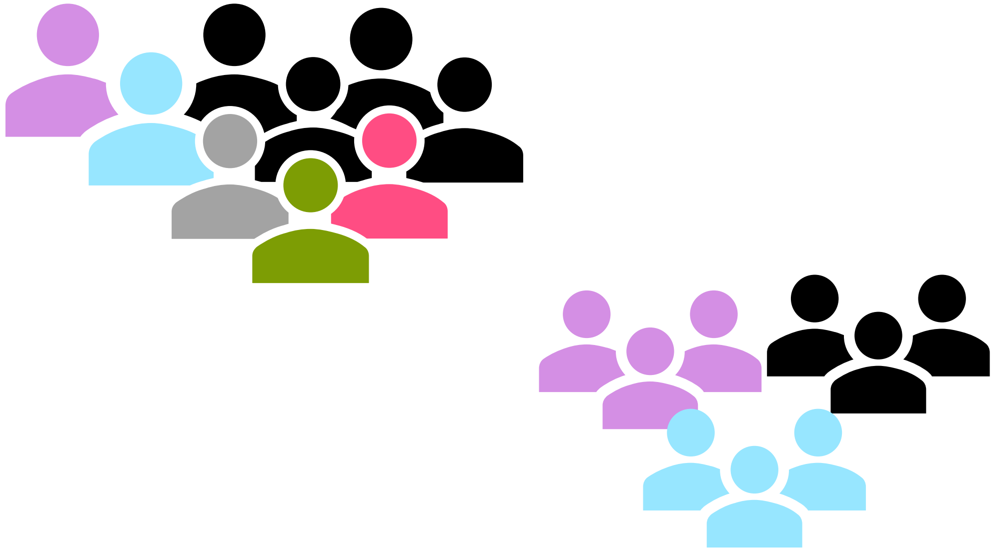
One partner requirement area

- Sprint Planning 1 and Sprint Planning 2
  - → within each area independently → okay

- Sprint and common code base
  - okay

Mix of partners

Mix of partners



- Extremely complicated situation
  - Are the different partners allowed to work collaboratively?
  - Rules for communication?
  - Rules for aligning?
  - Multi-team … possible?
- How to contract, model and live this so that the legal audit “okay” this?
- Need to evaluate thoroughly if the system effect still desired

Watch critically for
DOs and DON’T DOs

valtech_

“Leading Team”

Standard LeSS

Leading Team

- Several aspects of a leading team
  - Venturing a new complex topic with monstrously large features and domains
  - “leading role for giant feature”
  - …

Leading Team

Leading role aspect with partners

- Coordinate with partners
  - Ideally no added waste in the process
    - No refinement
    - No code integration
  - But they may ease some legal issues
    - No shared code ownership, due to contractual or other legal consideration
      - E.g. integrate and then “own” the code
    - Consideration due to intellectual property
      - E.g. integrate and then “own” the code

05

section

Additional collaboration aspects

Additional collaboration aspects

- Traveler
- Communities
- Decisions
- Shared infrastructure
- Coaches
- “roaming” Scrum Masters

thank you