More with LeSS

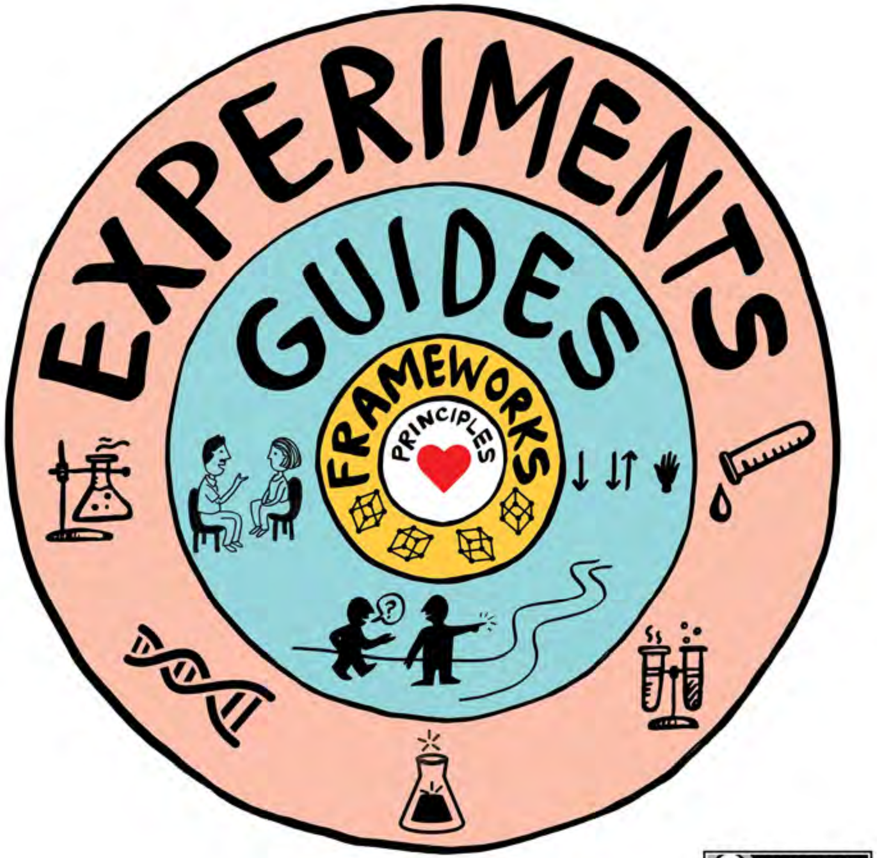
Presented by Richard Hundhausen

Agile Treasure Valley Meetup | 24 Mar 2016

Agile Scaling Frameworks

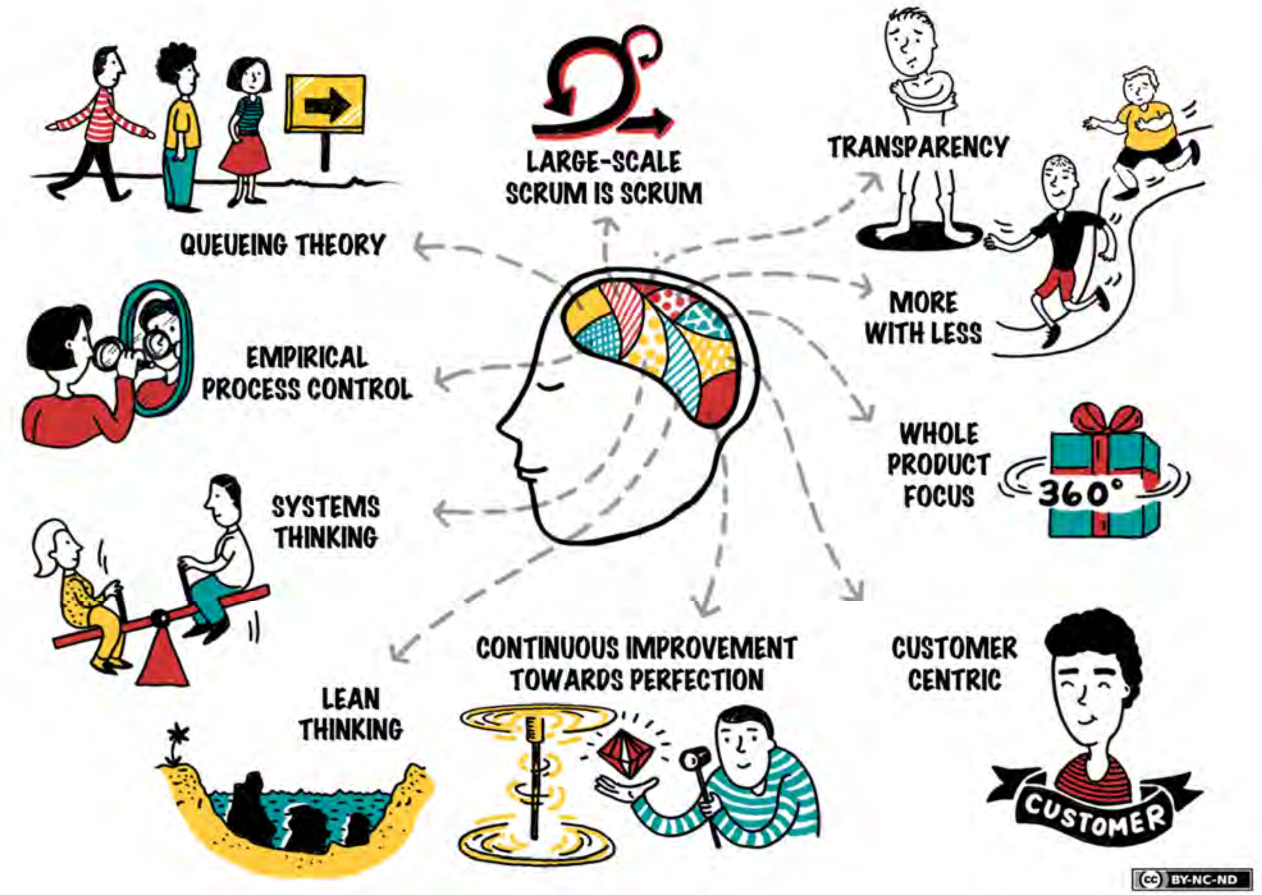
- Large Scale Scrum (LeSS)
- Nexus
- Scaled Agile Framework (SAFe)
- Scrum at Scale

What is LeSS?

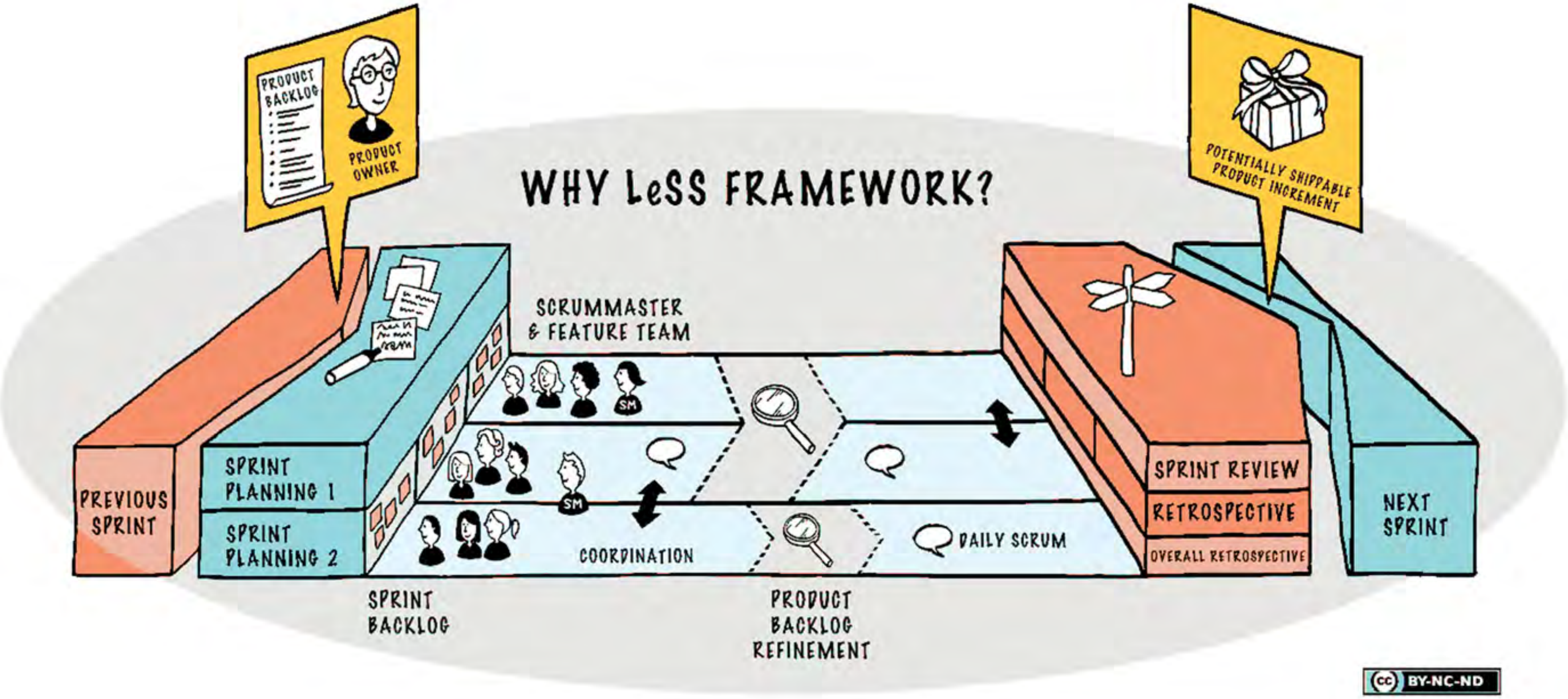


Created by These Guys

LeSS Principles



What is LeSS?



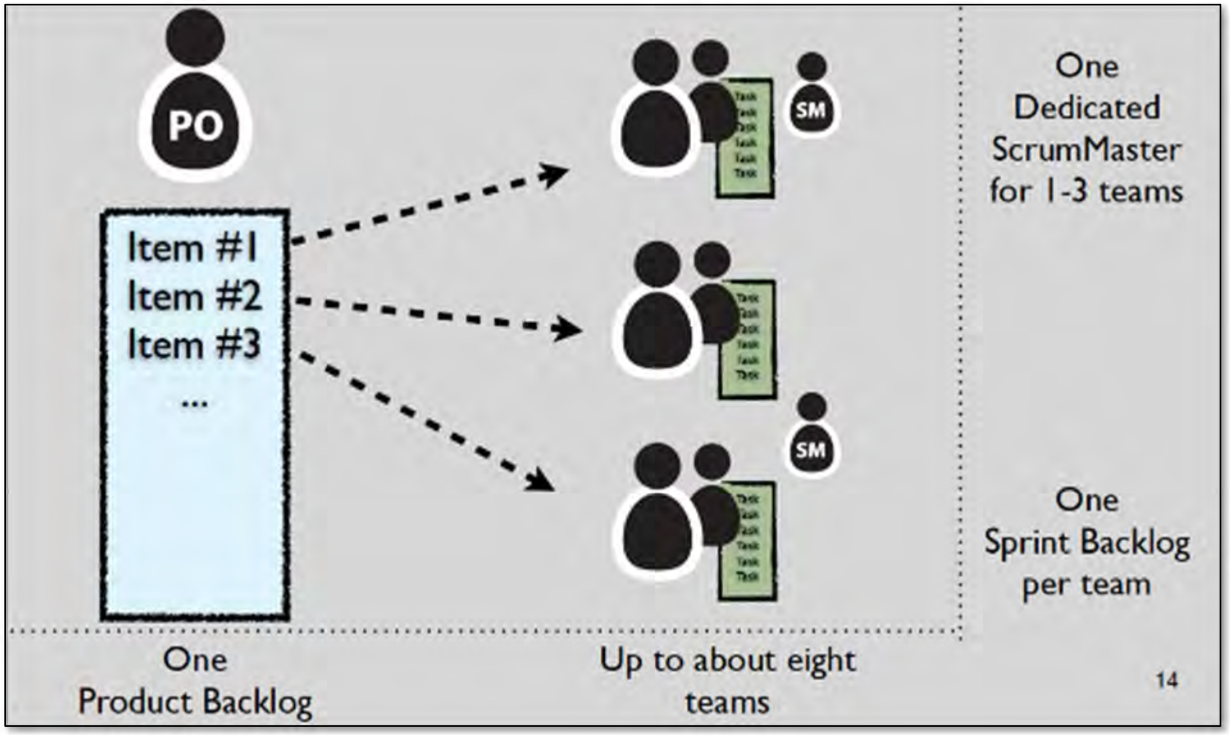
PRODUCT BACKLOG
PRODUCT OWNER
WHY LeSS FRAMEWORK?
POTENTIALLY SHIPPABLE PRODUCT INCREMENT
SCRUMMASTER & FEATURE TEAM
SM
PREVIOUS SPRINT
SPRINT PLANNING 1
SPRINT PLANNING 2
COORDINATION
DAILY SCRUM
SPRINT REVIEW
RETROSPECTIVE
OVERALL RETROSPECTIVE
NEXT SPRINT
SPRINT BACKLOG
PRODUCT BACKLOG REFINEMENT

Where's the Scrum in LeSS?

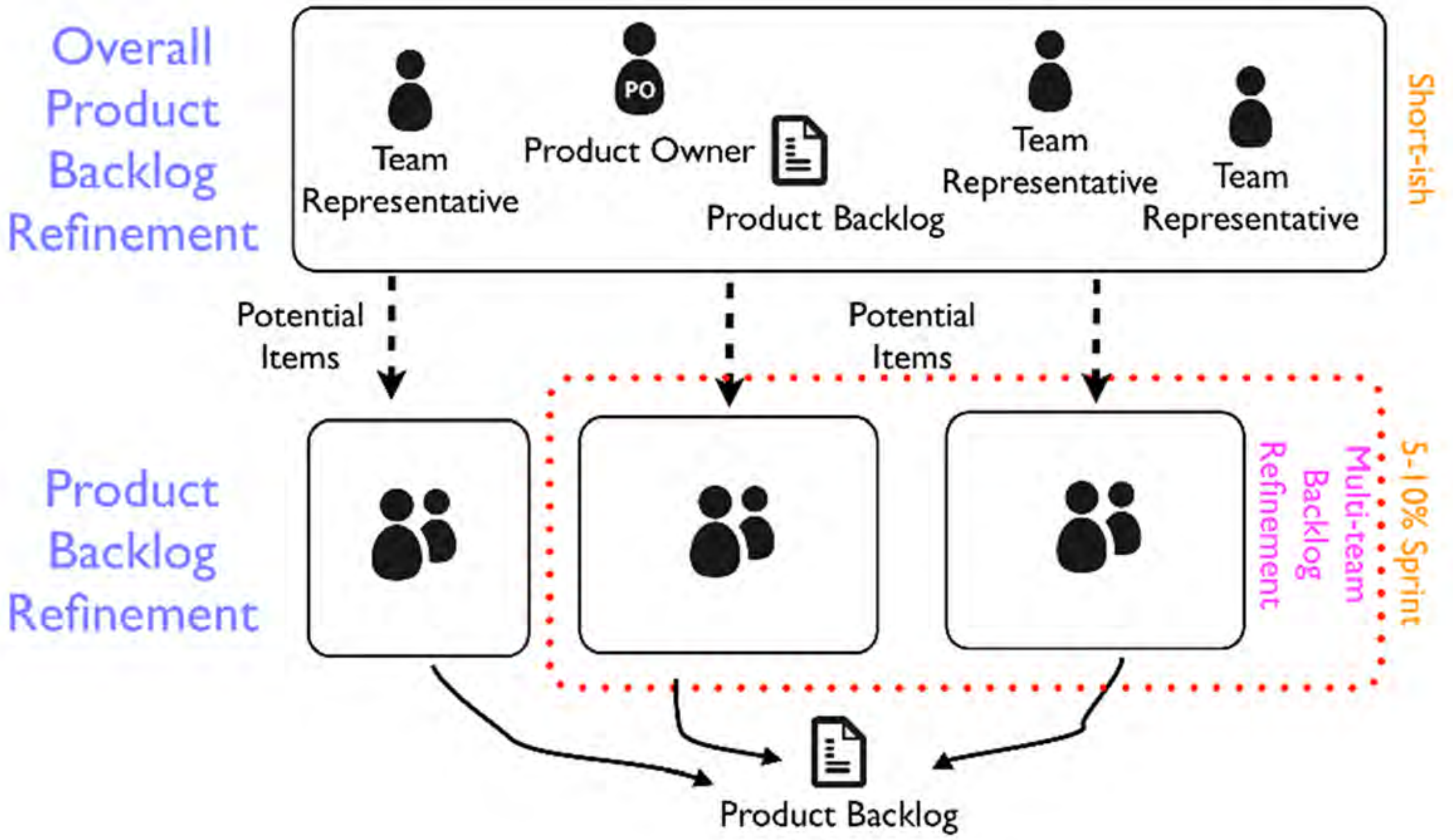
- Sprint Planning Part 1 & Part 2
- Daily Scrum
- Sprint Review
- Sprint Retrospective & Overall Retrospective
- Product Backlog Refinement

What is LeSS?

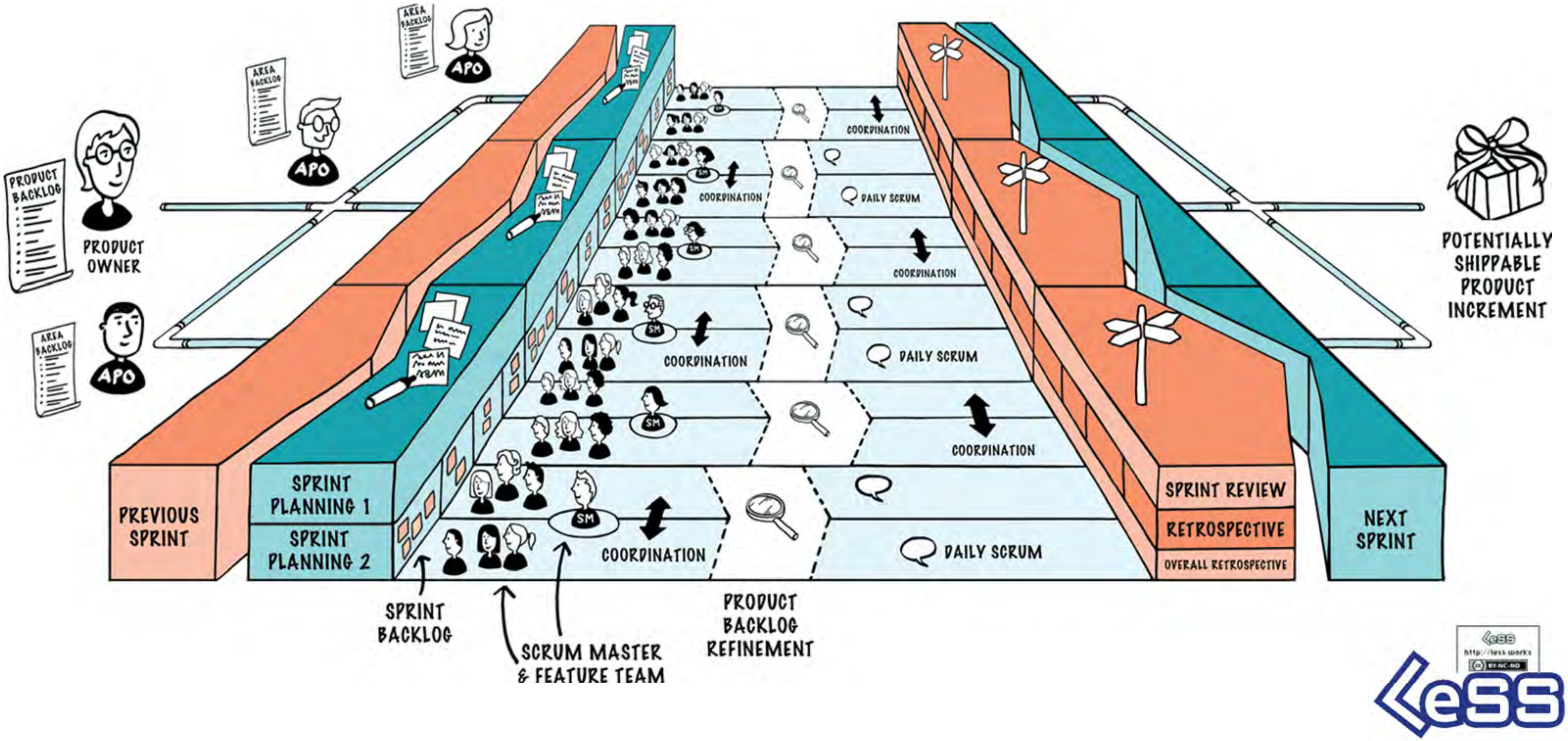
- 2-8 teams
- One Backlog / Product Owner



Backlog Refinement

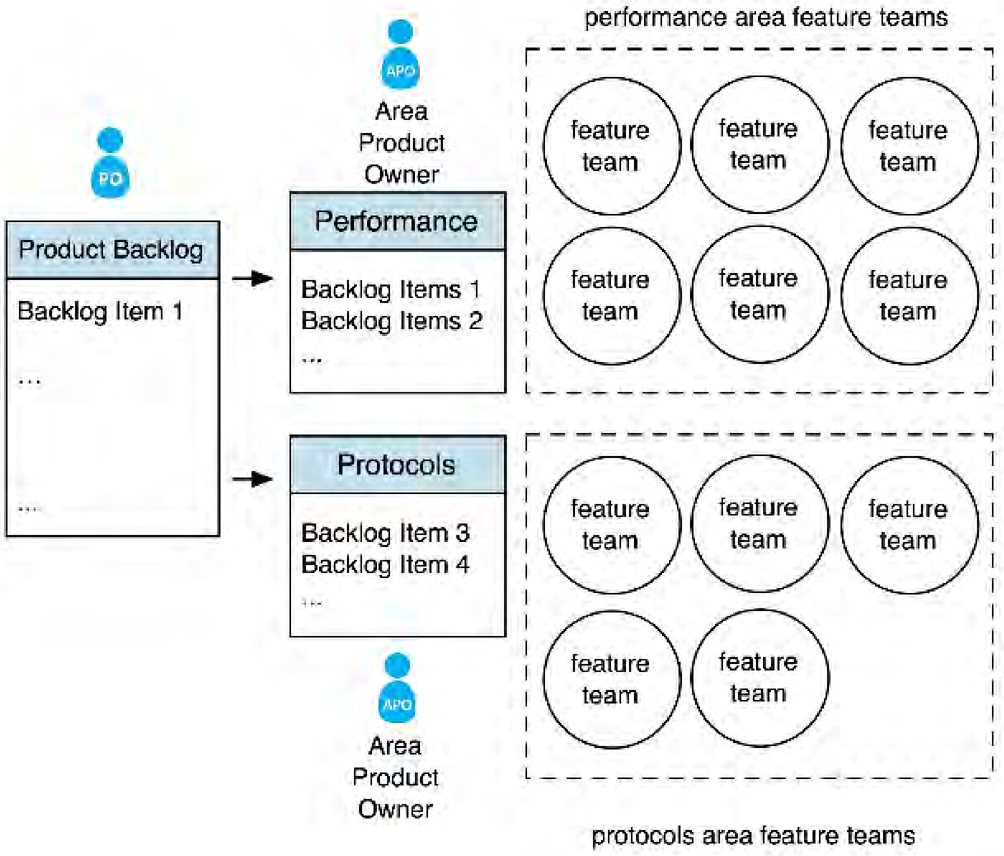


LeSS Huge?


AREA BACKLOG
APO
AREA BACKLOG
APO
PRODUCT BACKLOG
PRODUCT OWNER
AREA BACKLOG
APO
PREVIOUS SPRINT
SPRINT PLANNING 1
SPRINT PLANNING 2
SM
COORDINATION
DAILY SCRUM
SPRINT REVIEW
RETROSPECTIVE
OVERALL RETROSPECTIVE
NEXT SPRINT
POTENTIALLY SHIPPABLE PRODUCT INCREMENT
SPRINT BACKLOG
SCRUM MASTER & FEATURE TEAM
PRODUCT BACKLOG REFINEMENT
LeSS
http://less.works
BY-NC-ND

LeSS Huge?

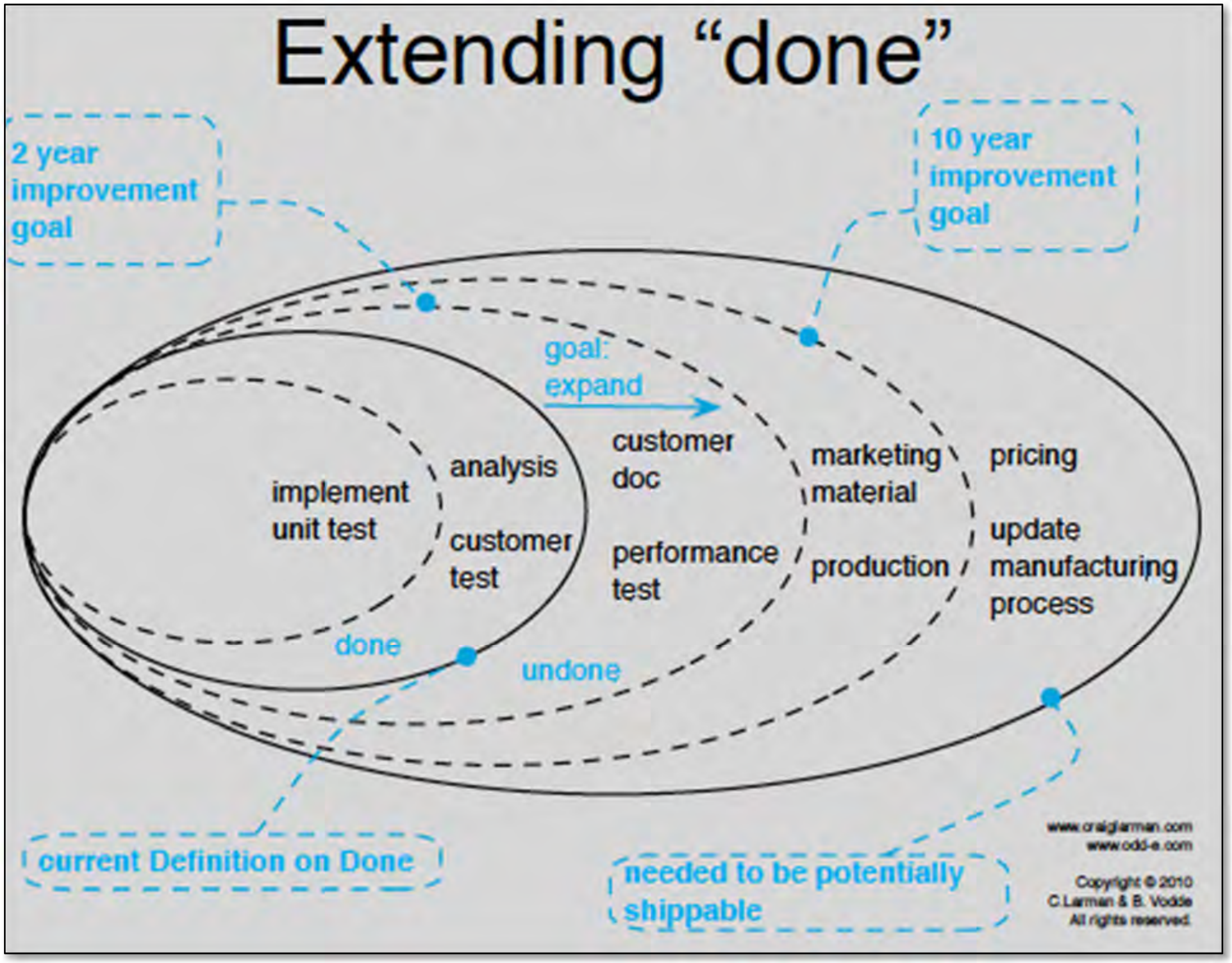
- 8 teams
- Product Areas
  - These are dynamic
- Area Product Owners
  - Act as PO toward team
  - Form the PO team



LeSS Concepts

- Build Your Method Up, Don’t Tailor It Down
  - “Barely Sufficient”
- Potentially shippable increment

LeSS Concepts



Extending “done”

2 year improvement goal

10 year improvement goal

goal: expand

implement unit test

analysis

customer test

customer doc

performance test

marketing material

production

pricing

update manufacturing process

done

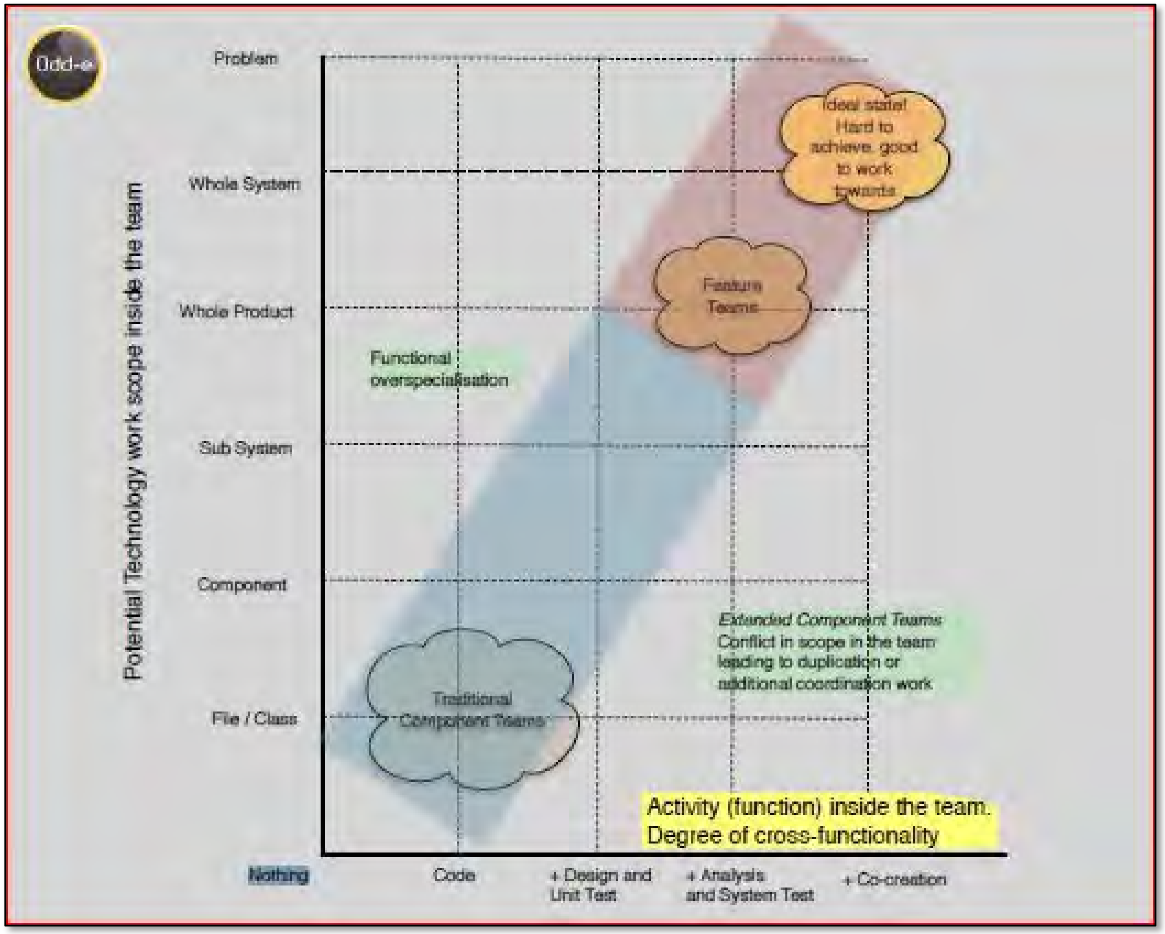
undone

current Definition on Done

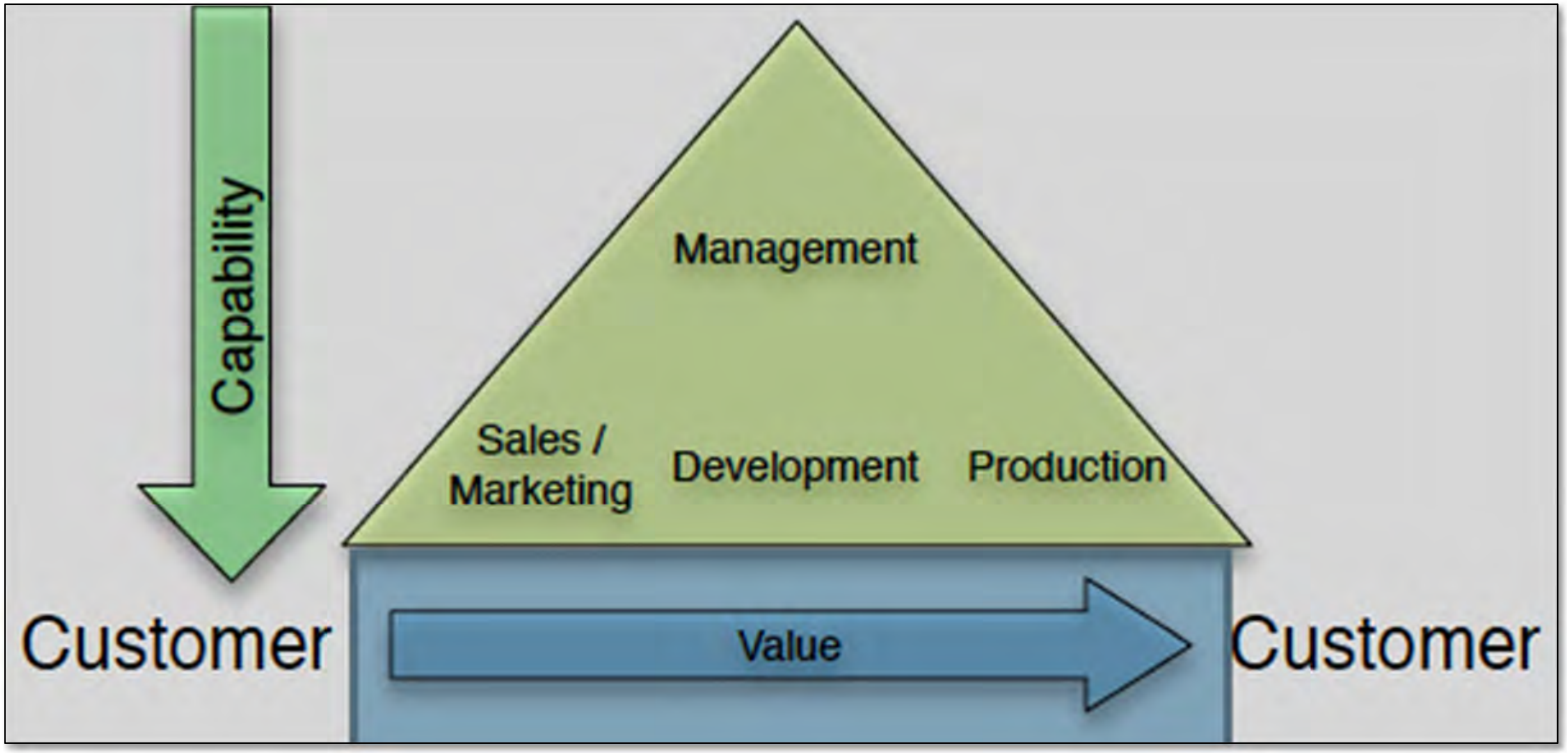
needed to be potentially shippable

www.craiglarman.com
www.odd-e.com

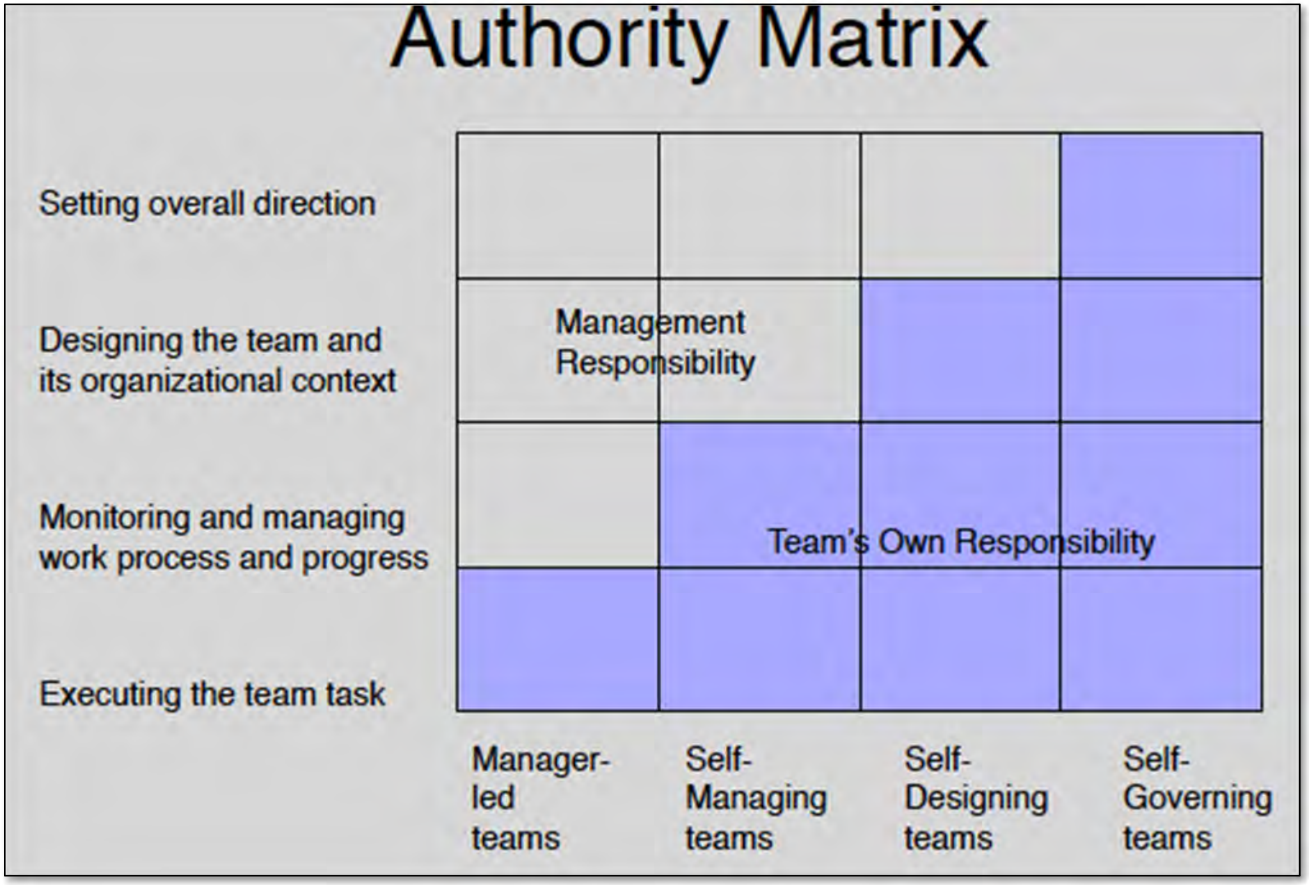
Generic Feature Team Map


Odd-e
Potential Technology work scope inside the team
Problem
Whole System
Whole Product
Sub System
Component
File / Class
Nothing
Code
+ Design and Unit Test
+ Analysis and System Test
+ Co-creation
Ideal state! Hard to achieve, good to work towards
Feature Teams
Functional overspecialisation
Traditional Component Teams
Extended Component Teams
Conflict in scope in the team leading to duplication or additional coordination work
Activity (function) inside the team.
Degree of cross-functionality

Value & Capability



LeSS Concepts



Authority Matrix

	Manager-led teams	Self-Managing teams	Self-Designing teams	Self-Governing teams
Setting overall direction				▨
Designing the team and its organizational context			▨	▨
Monitoring and managing work process and progress		▨	▨	▨
Executing the team task	▨	▨	▨	▨

Management Responsibility

Team's Own Responsibility

LeSS Benefits/Strengths

- Light framework
- Very scrum-like
- Addresses issues with organization and manager roles in a way that hasn't been addressed by non scaling agile frameworks (including Scrum)
  - Moves roles like Business Analyst and Architect on to the team
- Structured to facilitate team self management
  - Sprint goal not required

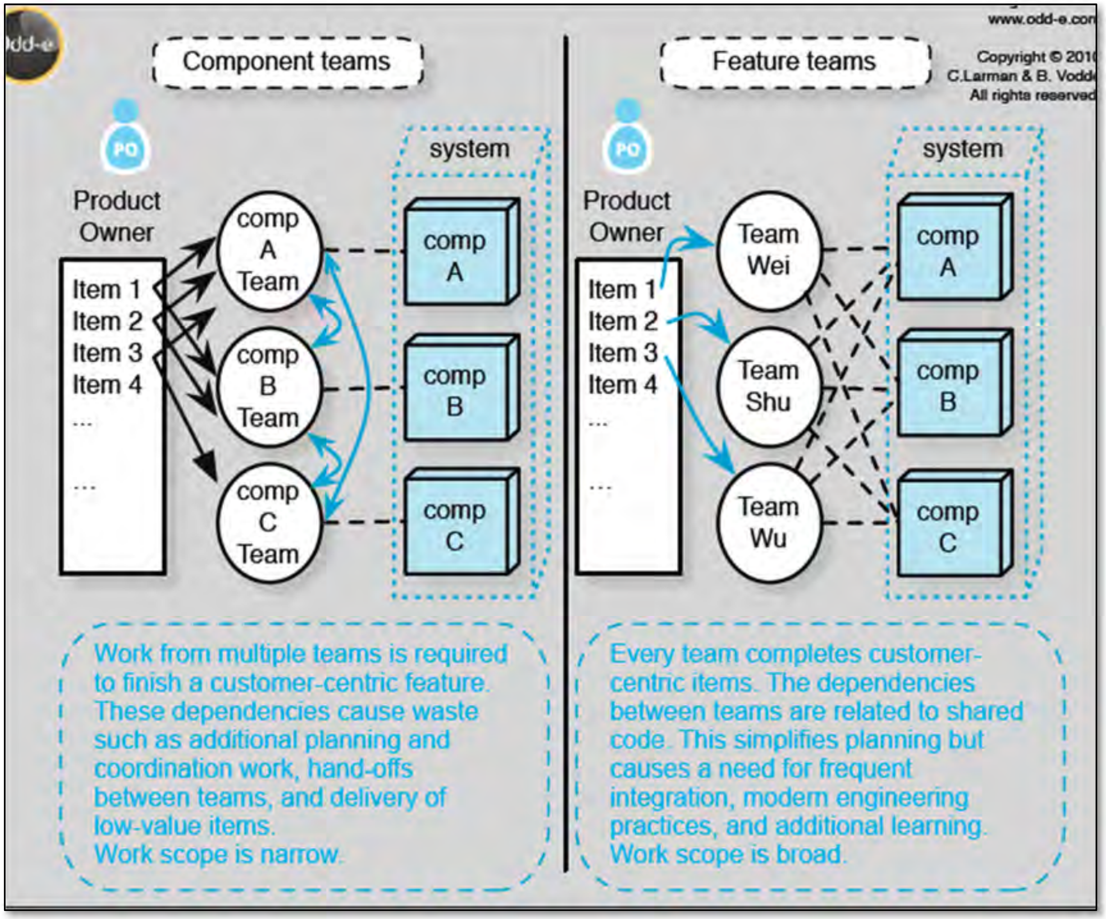
LeSS Benefits/Strengths

- Cuts down on dependency delays by creating feature teams
  - Tries to push dependencies to code level that are worked out by development teams
  - Coordination matches integration; use "social coding" tools like github/gitlab
  - Multi-team backlog refinement (2 members refine items 1-4, 2 refine items 5-8, etc.)

Team-Based Organizations

- Dedicated teams
- Cross-functional teams
- Co-located teams
- Long-lived teams
- Feature teams

Component v. Feature Teams



Work from multiple teams is required to finish a customer-centric feature. These dependencies cause waste such as additional planning and coordination work, hand-offs between teams, and delivery of low-value items.
Work scope is narrow.

Every team completes customer-centric items. The dependencies between teams are related to shared code. This simplifies planning but causes a need for frequent integration, modern engineering practices, and additional learning.
Work scope is broad.

LeSS Benefits/Strengths

- Encourages relationships/communication between stakeholders and teams
- A way of creating organizational change that is logical and makes sense
- Changes the structure of Management: each bus. unit has a requirement area manager to whom the feature teams report

LeSS Benefits/Strengths

- Cross-functional teams associated with a product owner and no departmental managers (e.g., development, QA) avoid conflict found between functional teams with separate managers
- Very clear path to focus teams and organization on producing customer value – rather than keeping people busy
- One product owner for many teams – creates independent teams

LeSS Benefits/Strengths

- One Scrum Master for 1-3 teams creates a scenario where the Scrum Master is really a coach, not a team admin; team has to manage itself
- Management or product now is intrinsically involved in what's being built; everyone is on the same team, so no us-versus-them between Development and Product

LeSS Difficulties

- Requires organization change from functional- to product- based (including reporting)
- Requires team members be co-located
- Requires all teams under one product use the same scale for estimation, which can limit effectiveness of relative sizing

LeSS Difficulties

- Structure – many companies are not ready for this
- Requirements of stakeholders to be involved for team can cause problems if they aren’t involved or available – could be a blocker