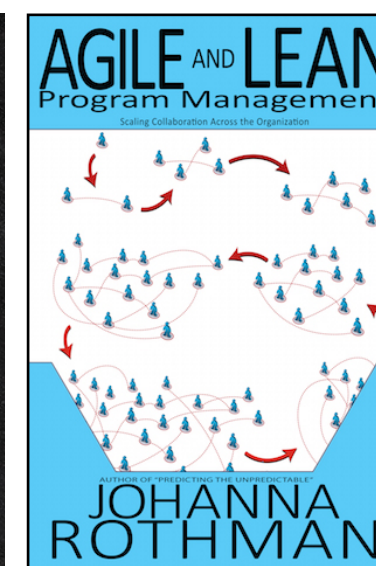
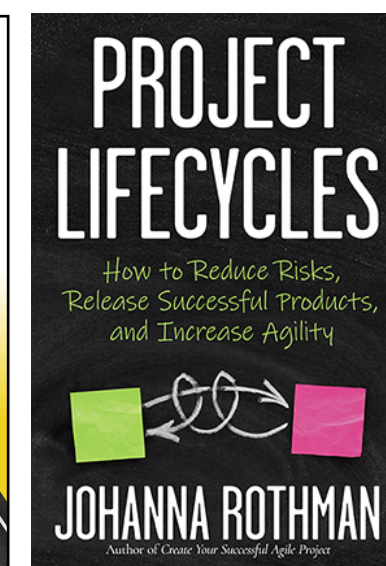
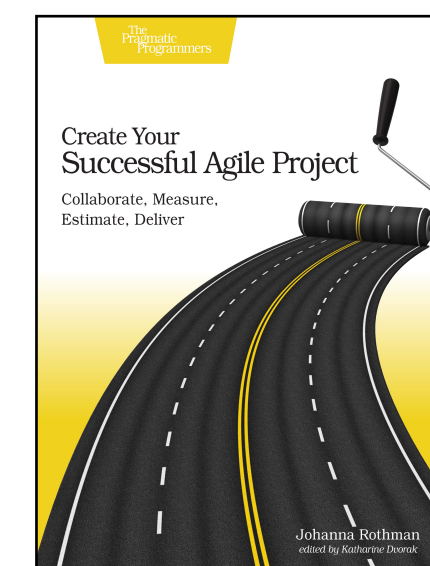
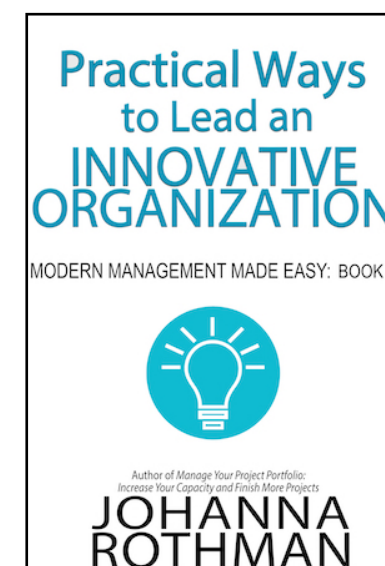
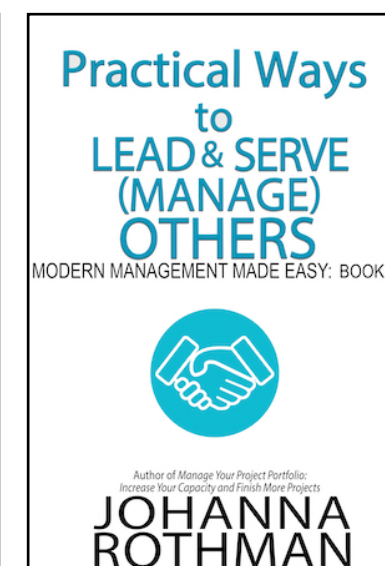
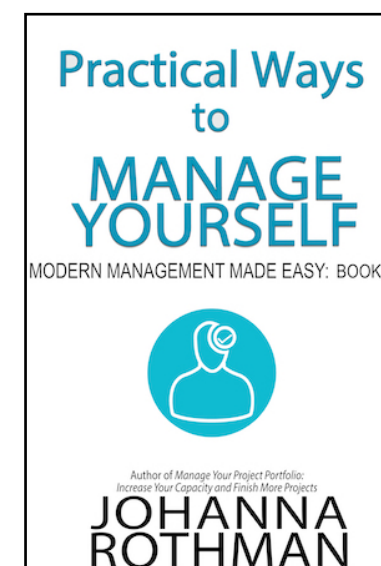


How to Use Value Stream Maps to Reinforce Agility and Effectiveness

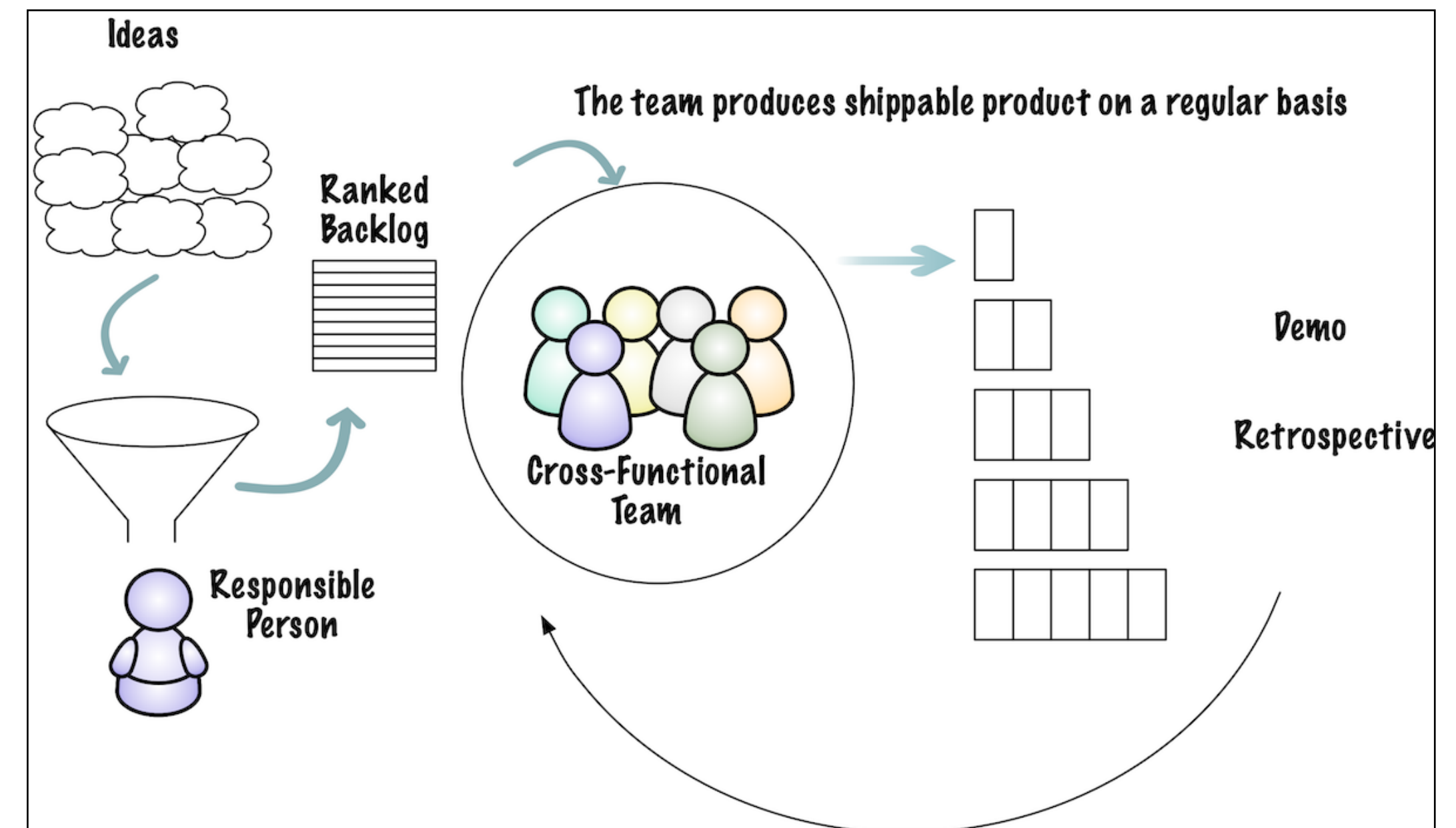
Johanna Rothman
jr@jrothman.com
www.jrothman.com



If we're so "agile," why does everything take so long??

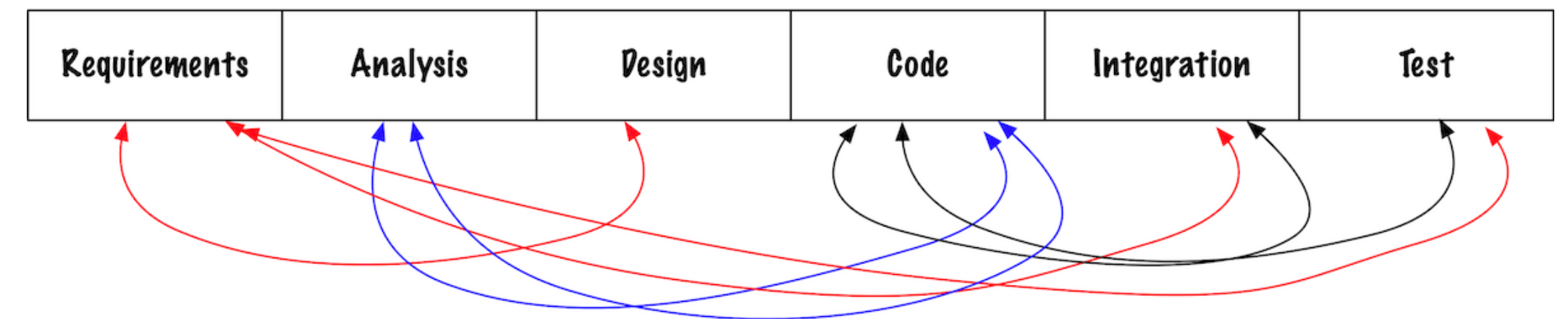
Agility Requires

- A collaborative, cross-functional team
 - Who takes work off a short, ranked backlog
 - Releases increments of value regularly
 - Demos regularly
 - Retrospects regularly
- All for the purpose of frequent-enough change



What I See is Not “Agile”

- Some common problems in too many “agile” teams:
 - Work rolls over from sprint to sprint
 - Testers (or some other group) cannot finish “on time”
 - The team has to wait for an expert or shared services team to be available
- The feedback loop from Ready to Demo is very long (unplanned feedback loops)



As a Result, Your Team Can't Predict Much

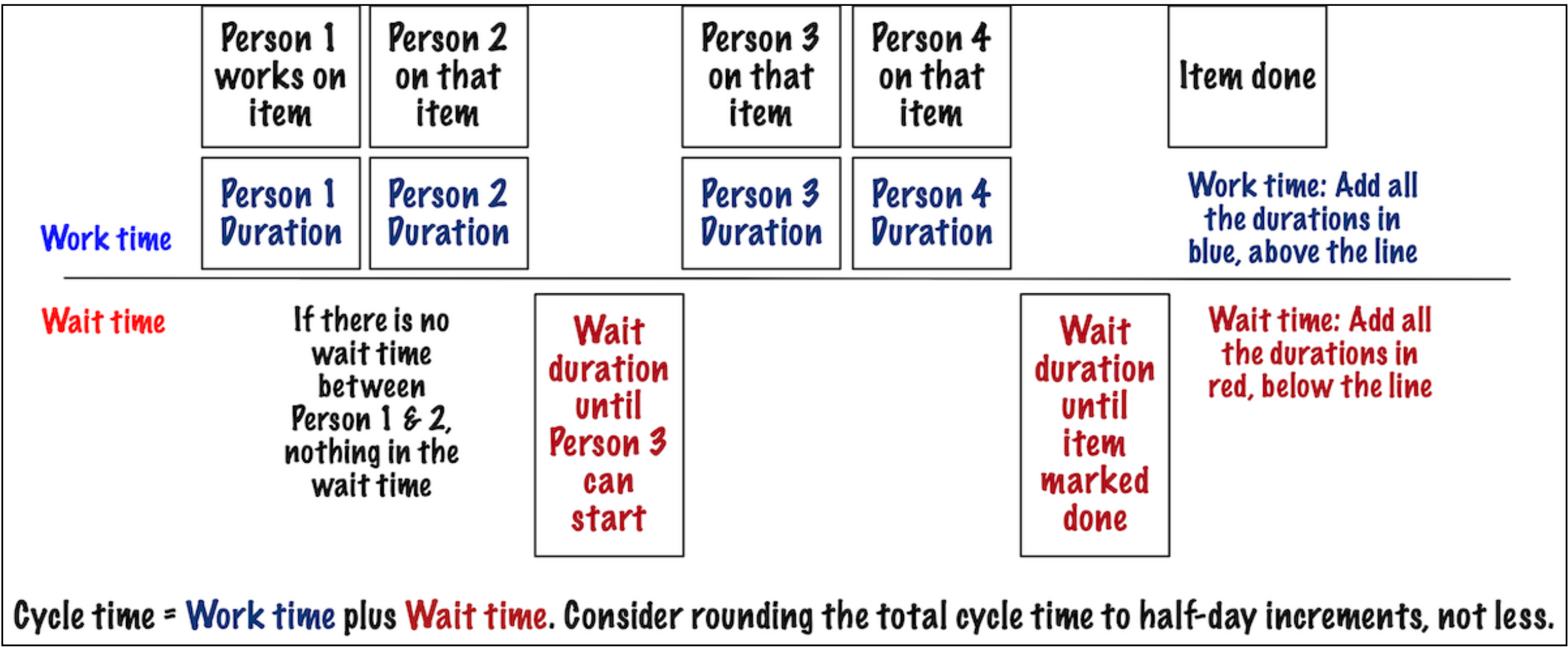
- Unreliable estimates (especially if you use story points)
- Frustration everywhere: teams and managers
- Because there is so little throughput, others ask people and teams to do more



Visualize the System of Our Work

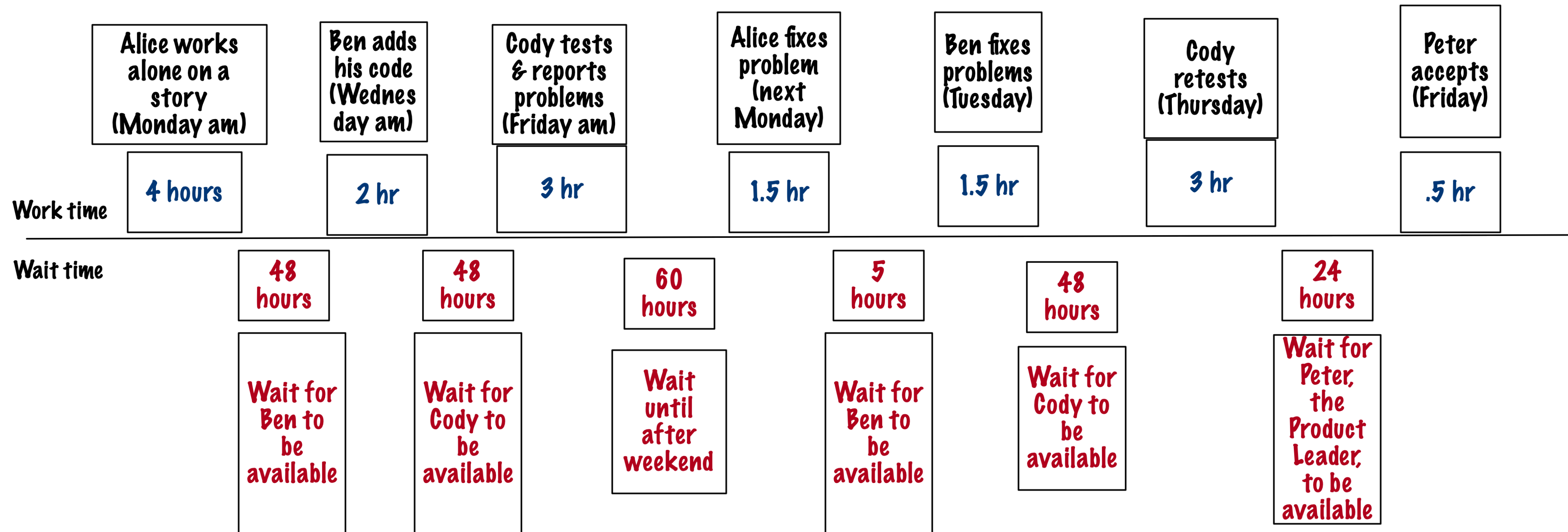
Value Stream Maps Show Everyone Your System

- Track the work time vs the wait time
- Each type of team has its own patterns:
 - Expert (component teams are a form of expert teams)
 - Cooperative
 - Collaborative



The Expert-Focused Value Stream Map

Three-Person "Team" Works as Experts : Value Stream Map for One Feature



Work time: $4 + 2 + 3 + 1.5 + 1.5 + 3 + .5 = 15.5$ hours

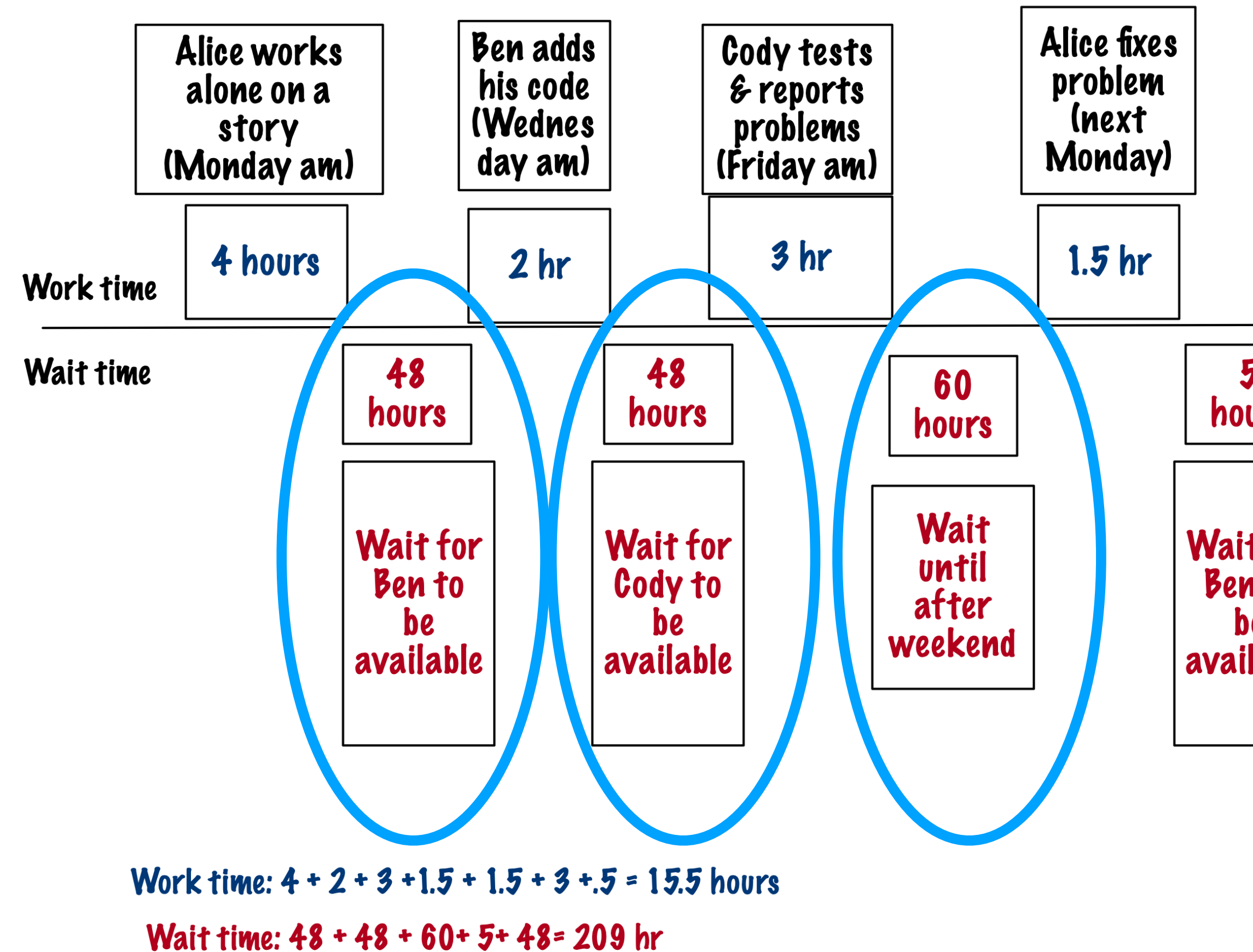
Wait time: $48 + 48 + 60 + 5 + 48 = 209$ hr

Cycle time = 15.5 plus 209 hr = 224.5 hr

Data from the Expert-Focused Value Stream Map

- Total cycle time of 224.5 hours with is 9.5 days (2 weeks)
- Work time was 15.5 hours (3 or 3+ days of work)
- Wait time is 209 hours, 8.7 days
- Also: what were people doing in all of that time?
 - Ask questions of others?
 - Service interruptions? (the longer the wait, the more likely there are other tasks)
 - Definitely multitasking during wait times

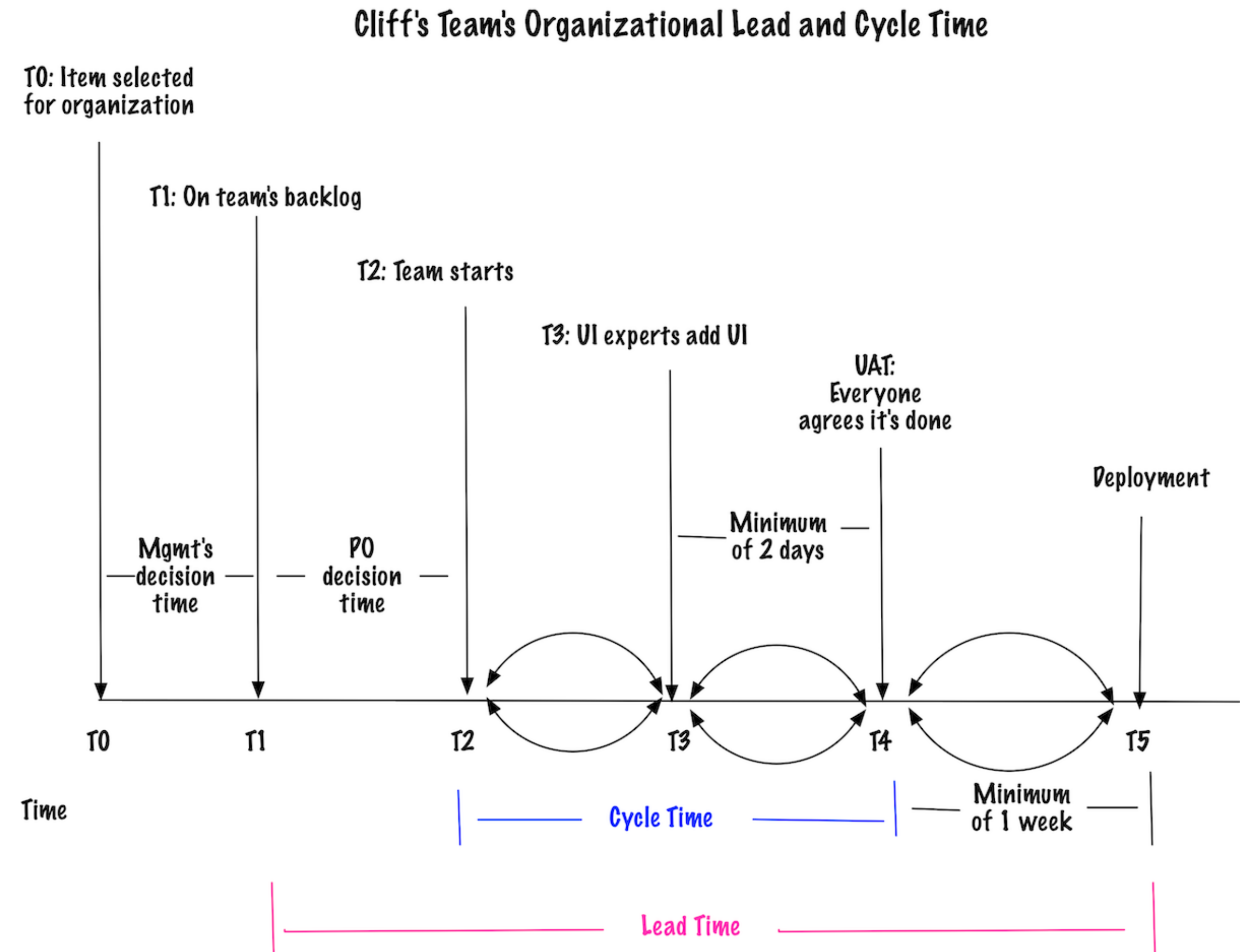
Three-Person "Team" Works as Experts : Value Stream Map



Cycle time = 15.5 plus 209

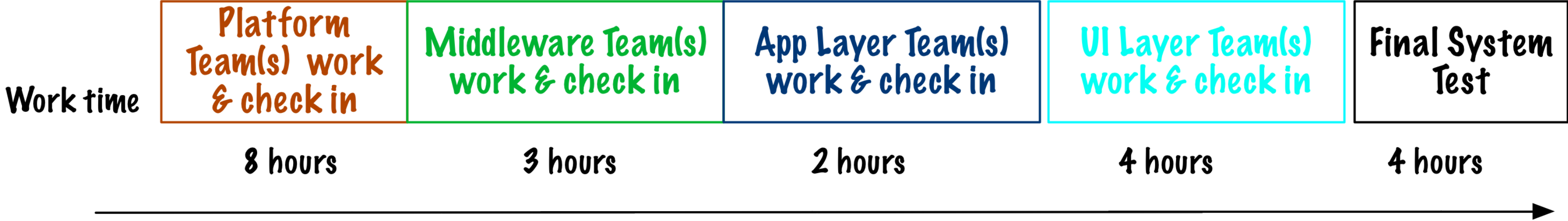
More Intentional Dependencies

- When orgs want to manage costs, they create “shared services”:
 - UI staff
 - UAT staff
 - Deployment
- In this org, story size was irrelevant. Every story had a cycle time of at least 2 (UI) + 5 (UAT) + 7 (Deployment) days: 14 days.



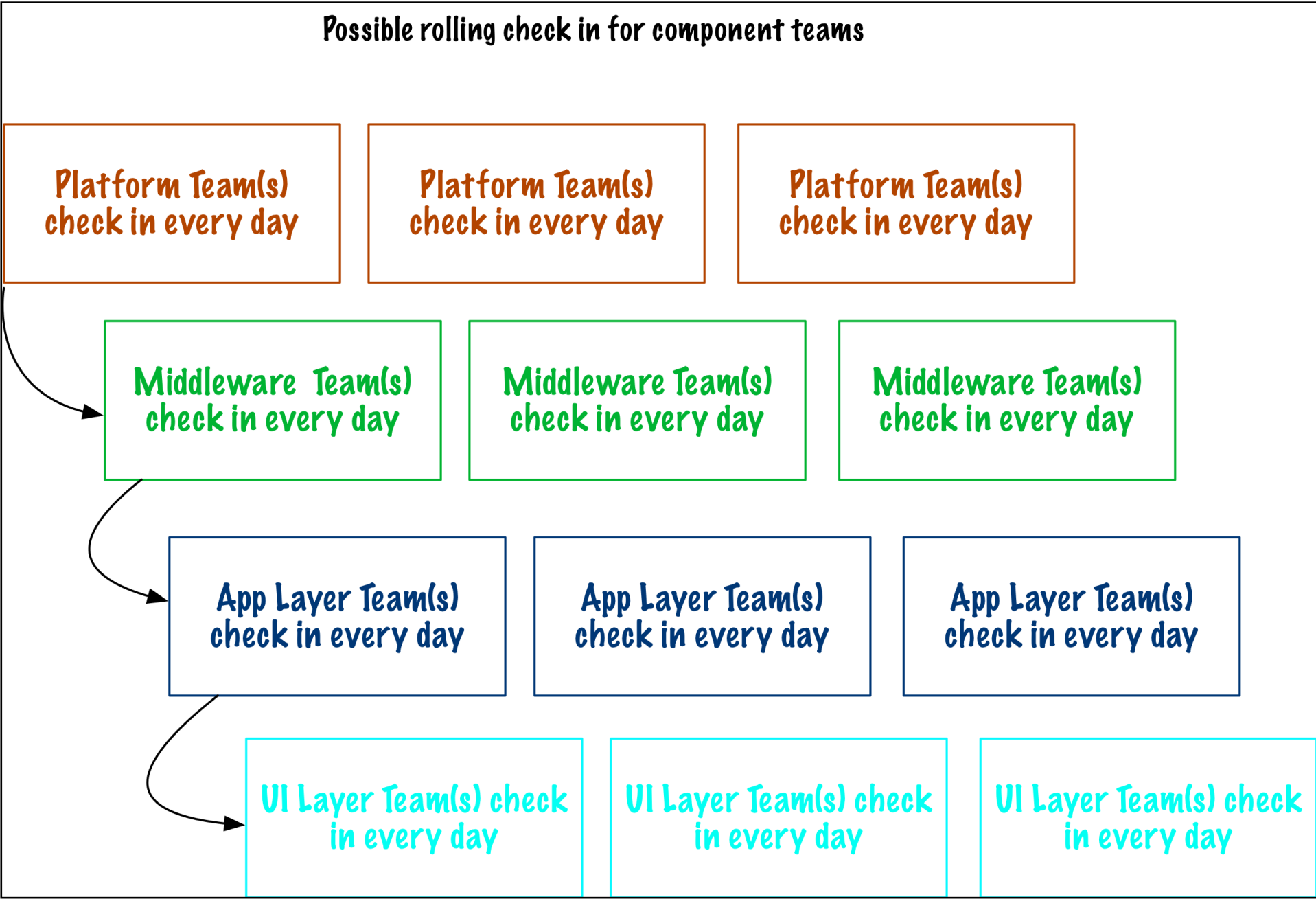
How Component Teams Can Work

Best Possible Cycle time for Component teams



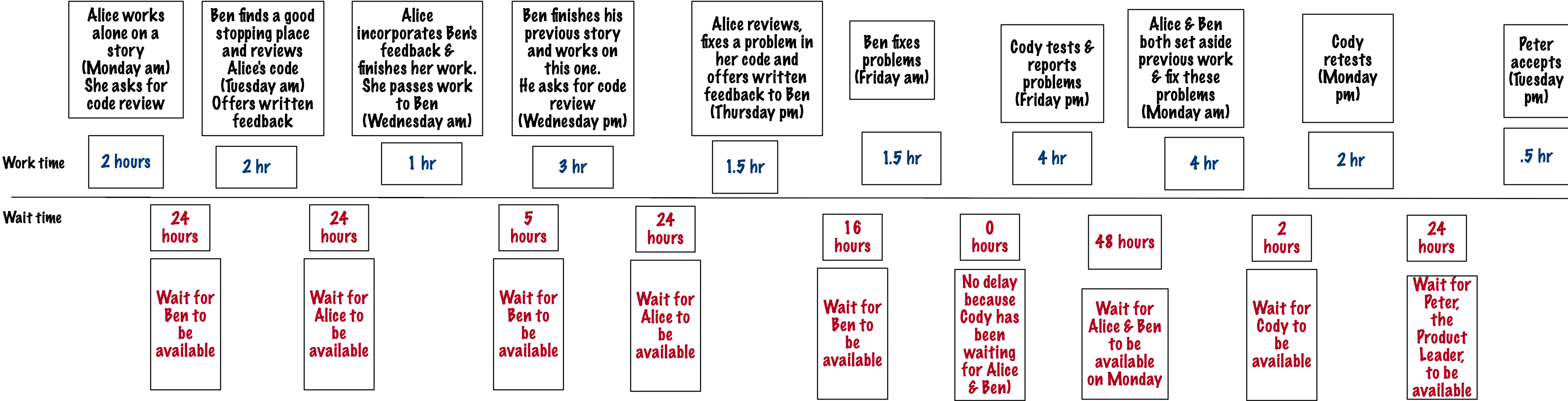
Wait time
Cycle time: 21 hours

One of my clients uses a daily rolling checkin for their component teams. That allows them to achieve a *best case* 21 hour cycle time.



Cooperative Team Value Stream Map

Three-Person "Team" Cooperates Using Working Agreements: Value Stream Map for One Feature



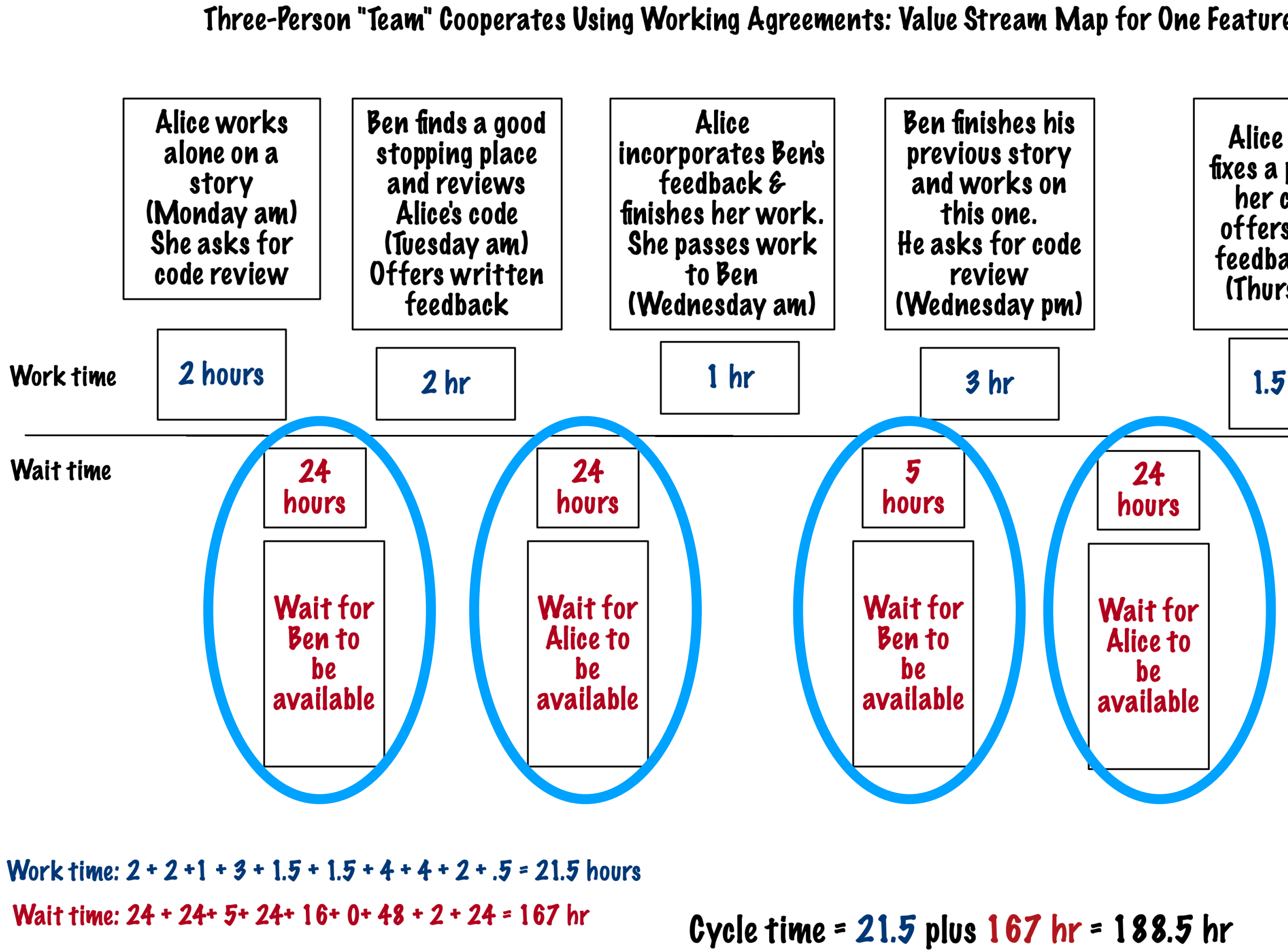
Work time: $2 + 2 + 1 + 3 + 1.5 + 1.5 + 4 + 4 + 2 + .5 = 21.5$ hours

Wait time: $24 + 24 + 5 + 24 + 16 + 0 + 48 + 2 + 24 = 167$ hr

Cycle time = 21.5 plus 167 hr = 188.5 hr

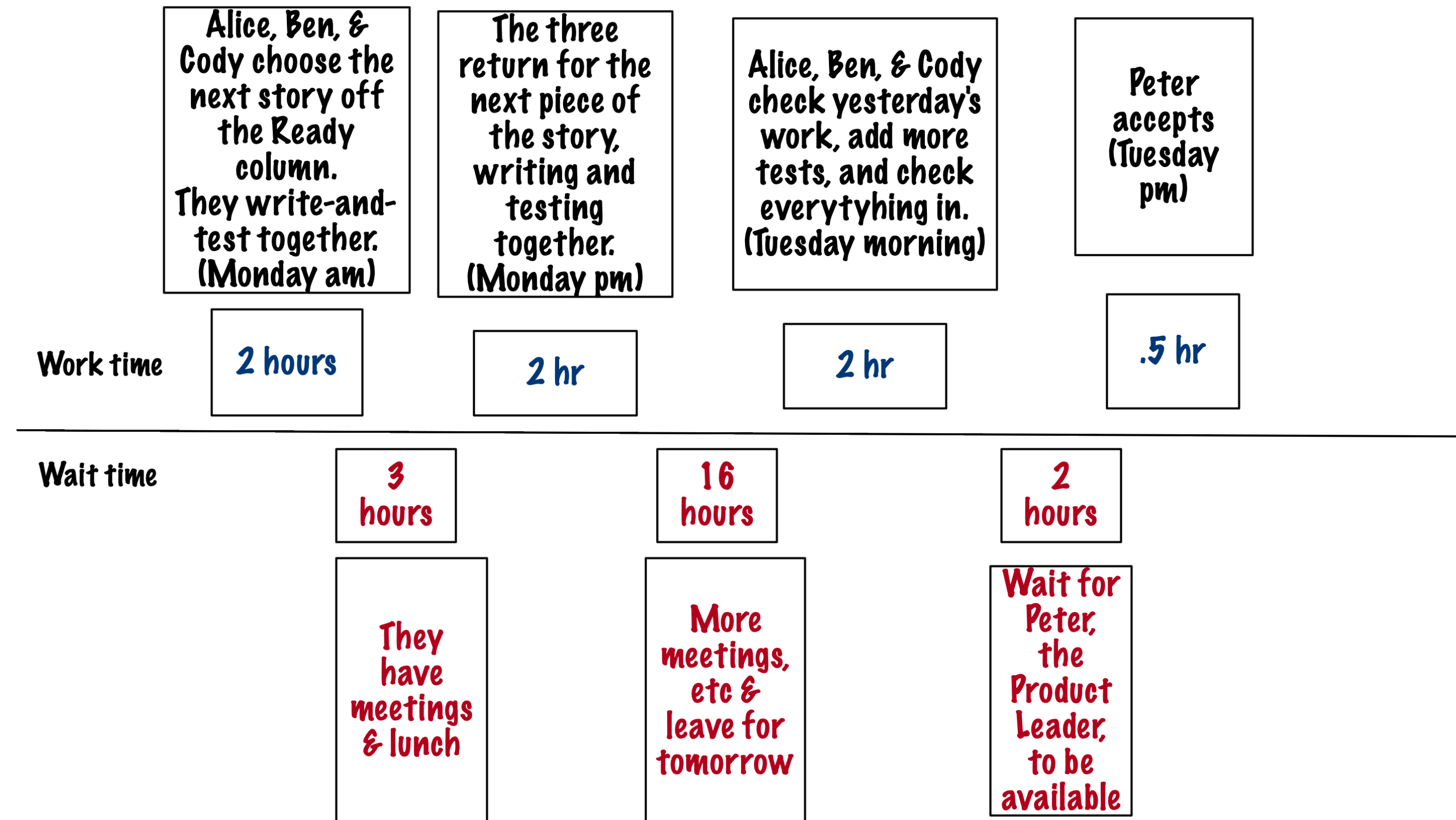
Data from the Cooperative Value Stream Map

- Total cycle time of 188.5 hours, almost 8 days.
- Work time was 21.5 hours (4 days of work)
- Wait time is 167 hours, almost 8 days
- What did people do in their wait times?
 - Ask or answer questions?
 - Service interruptions that are not stories?
 - Definitely multitasking



Collaborative Team Value Stream Map

Three-Person Collaborative Team



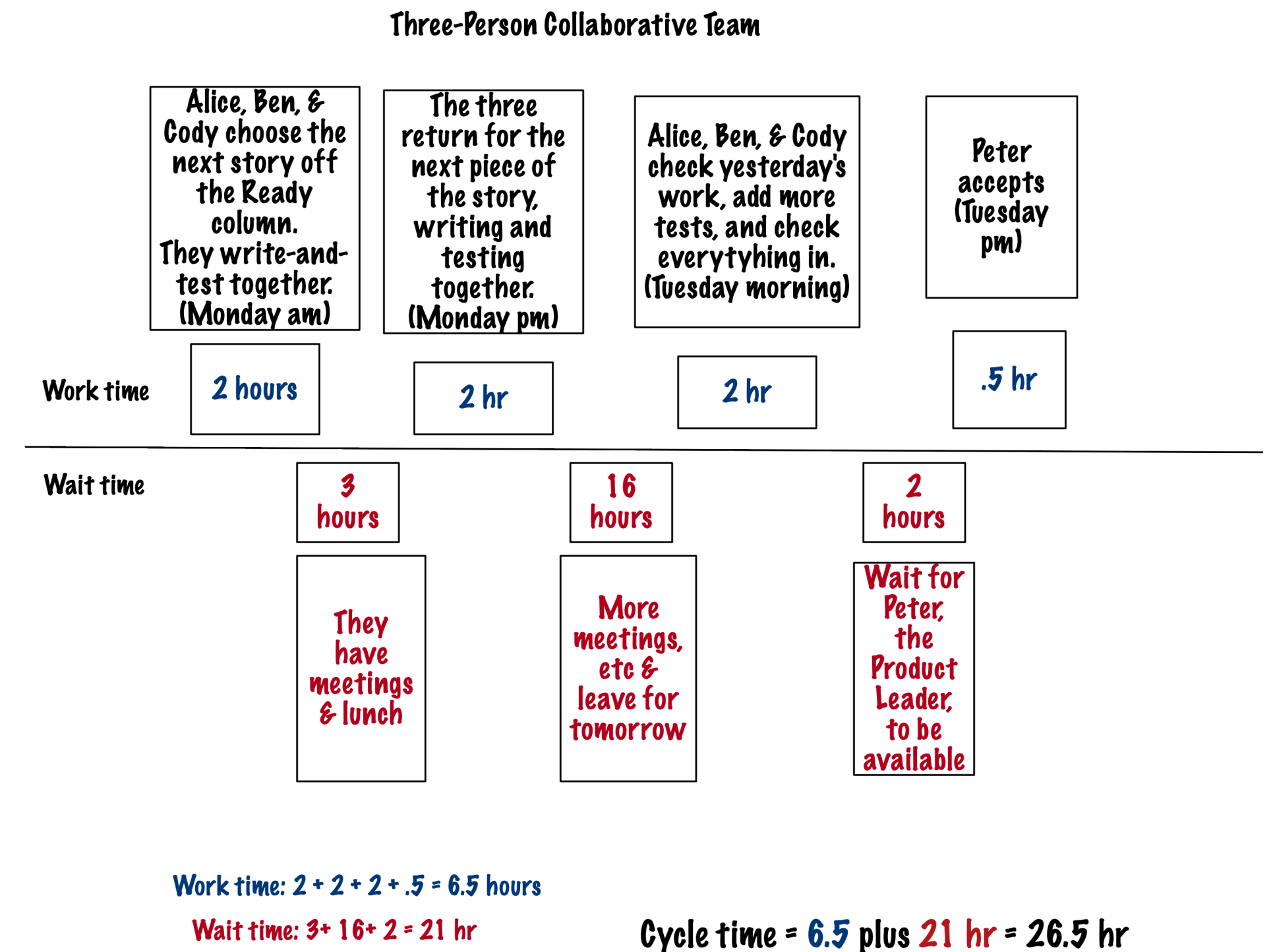
Work time: $2 + 2 + 2 + .5 = 6.5$ hours

Wait time: $3 + 16 + 2 = 21$ hr

Cycle time = 6.5 plus 21 hr = 26.5 hr

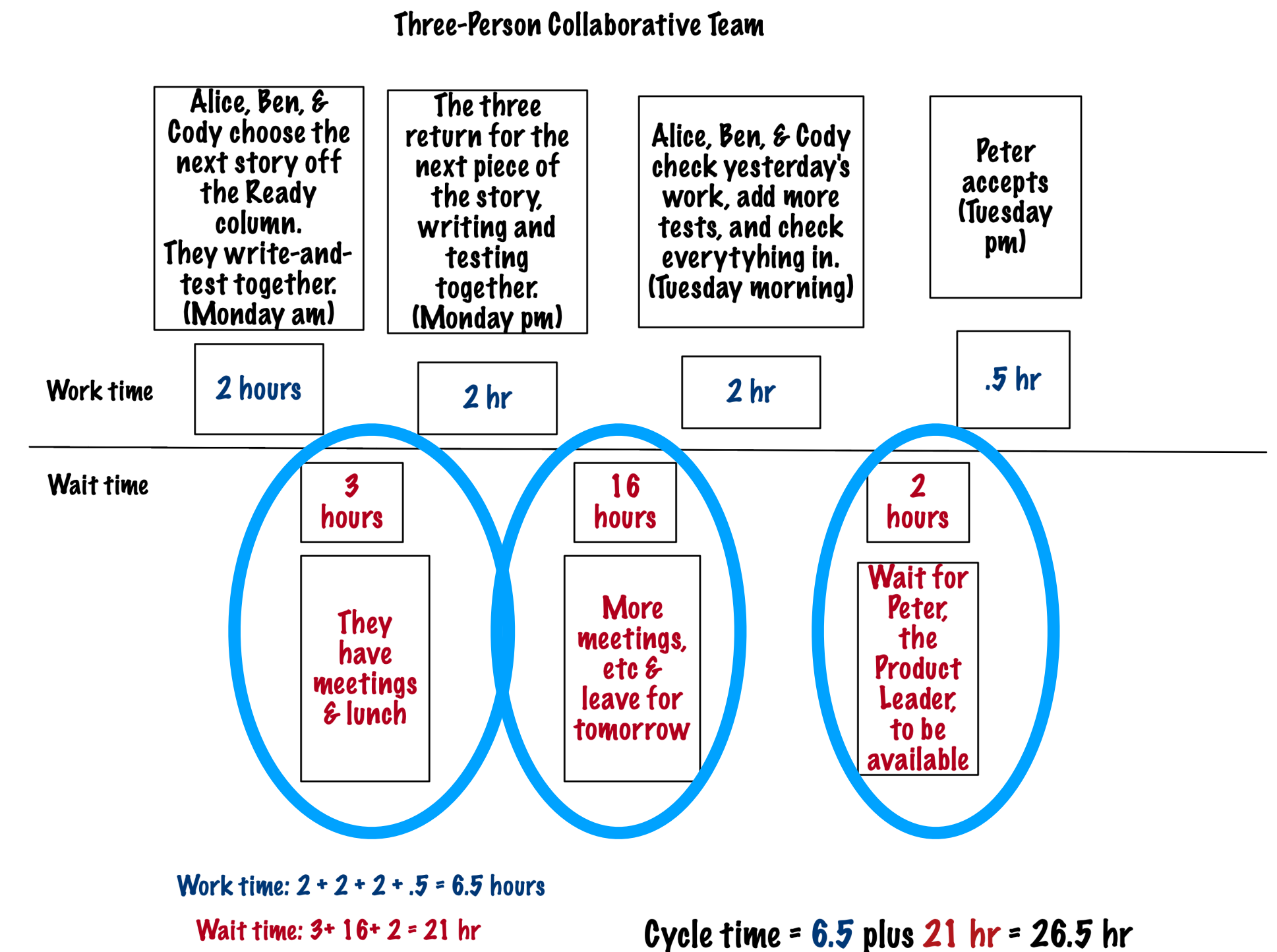
Data from the Collaborative Value Stream Map

- Total cycle time of 26.5 hours, just over a day
- Work time was 6.5 hours (1 day of work)
- Wait time is 21 hours, 1 day
- What did people do in their wait times?
 - The only *team* wait time was for Peter
 - They did not multitask or start new stories
 - A meeting-addicted organization



Collaborative Teams Exhibit Agility

- While this team had a lot of meetings (!), they were able to finish a story in a couple of days
- How many of you regularly have a cycle time of a day or two?



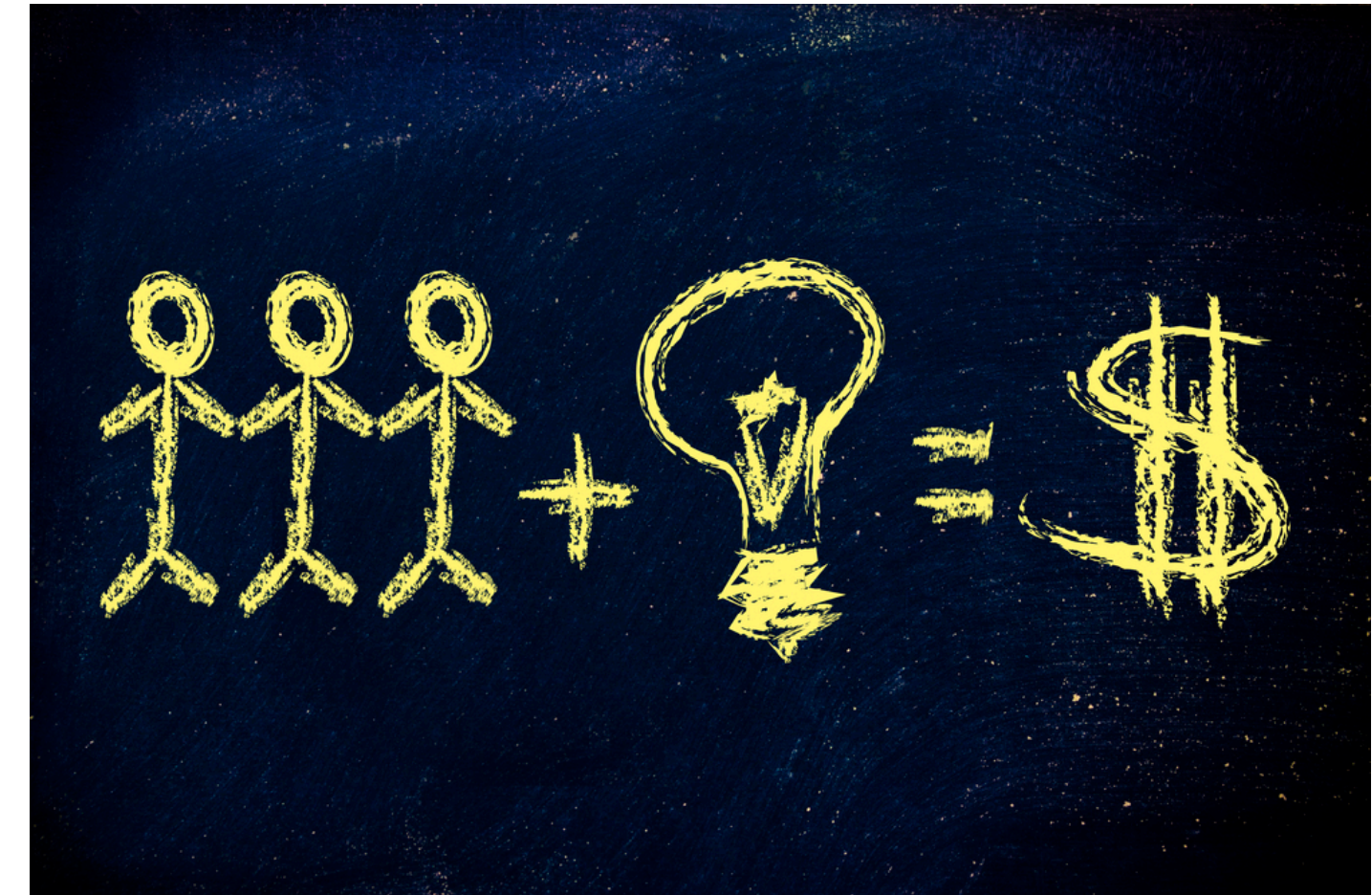
Several Collaboration Options

- Agility requires fast *team* learning
- That means the team must collaborate on limited WIP:
 - Pair: two people collaborate on one item: WIP of 1.
 - Swarm: Team or subset of a team collaborates on one item as experts: WIP is 1.
 - Mob: Entire team collaborates on one item: WIP is 1



What Does Management Want?

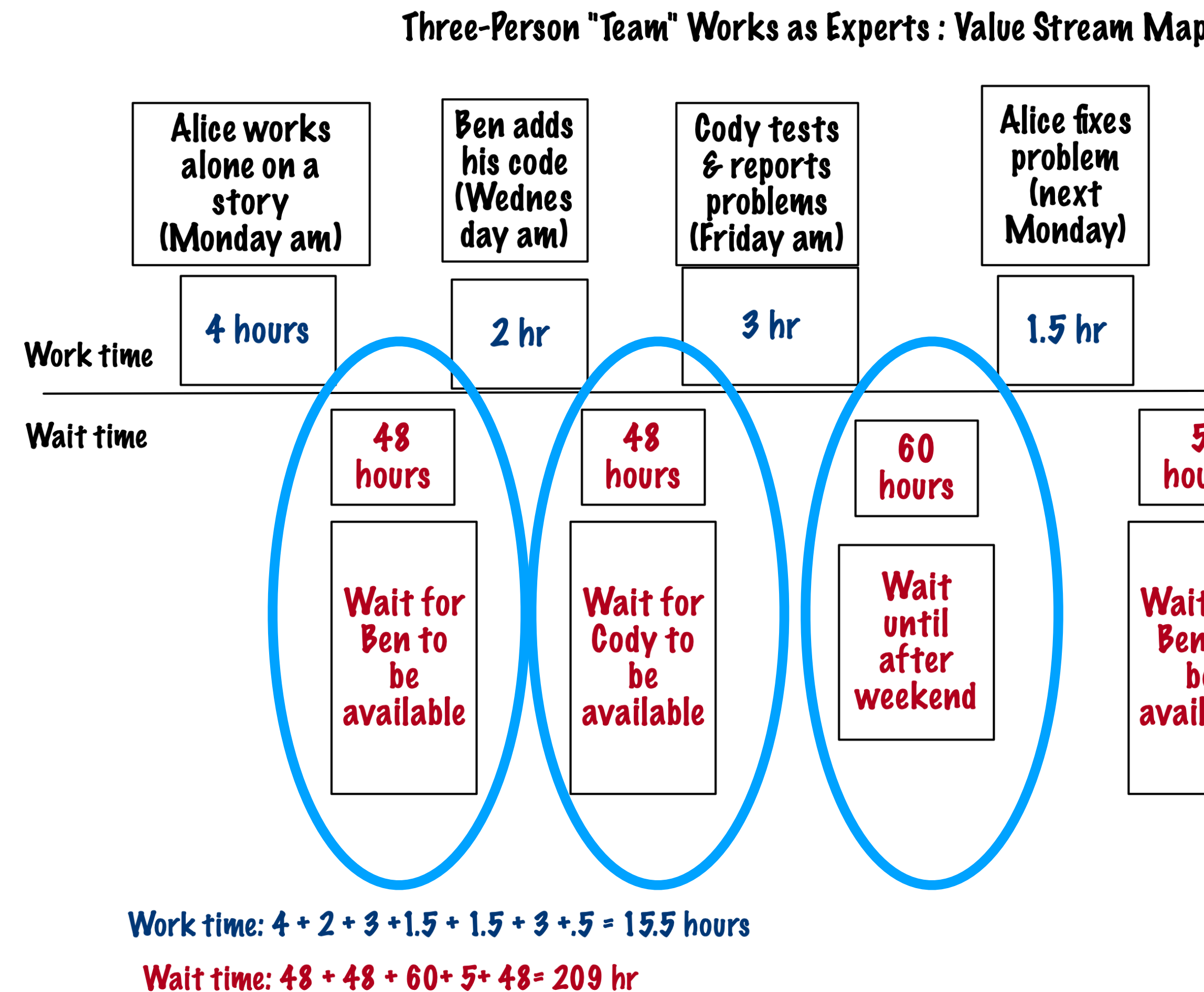
- Lower time to release
- Less investment
- That's why collaboration makes sense



**Long wait times do not offer what
management wants**

Ask Questions About Your Value Stream Maps

- Start with the *longest* wait times
- Context free questions often go “meta”: about the problem:
 - What problems does this delay solve elsewhere in the organization?
 - What problems does this delay create, not just in this project, but across the organization?
- (Avoid why questions because they sound blaming)

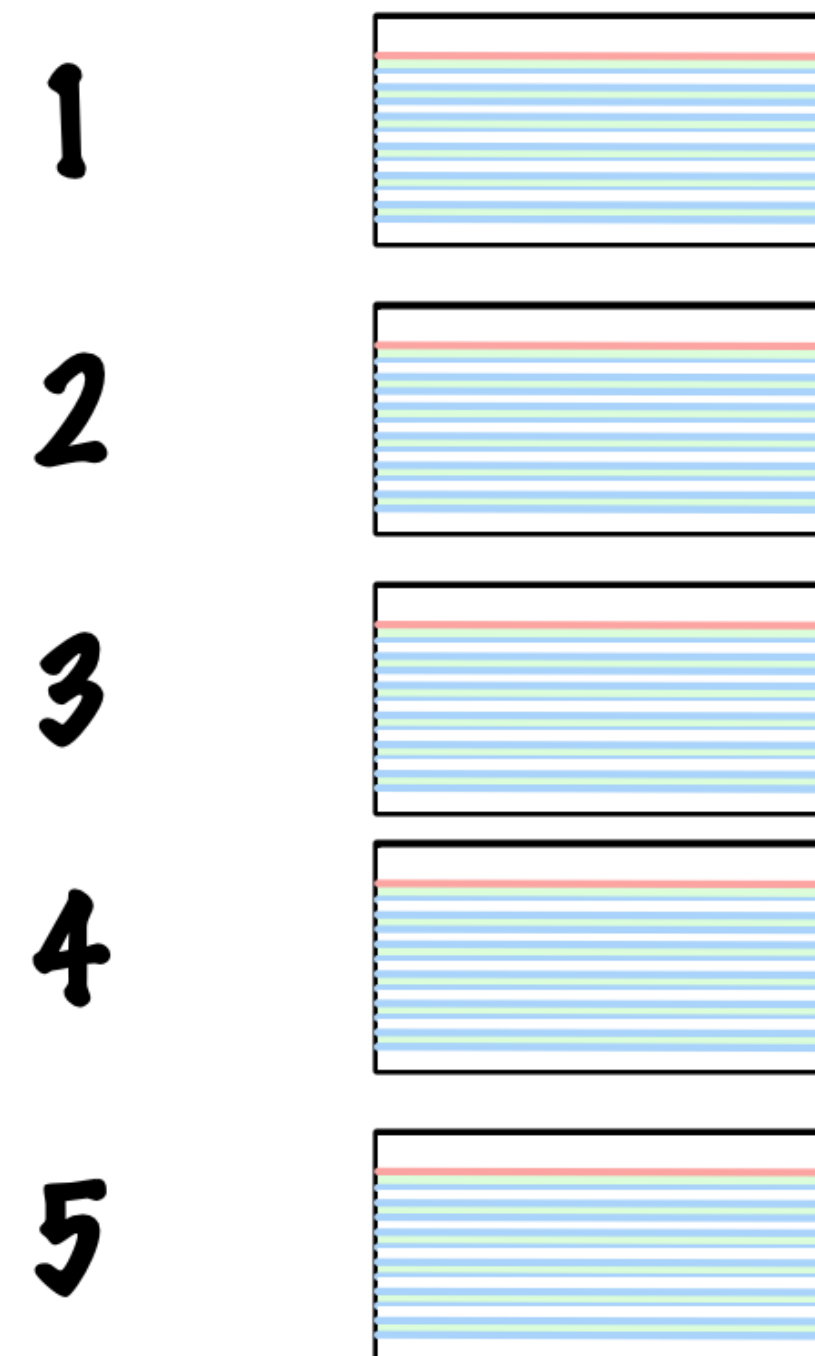


Cycle time = 15.5 plus 209

Mine Your Value Stream Maps for Objective Data

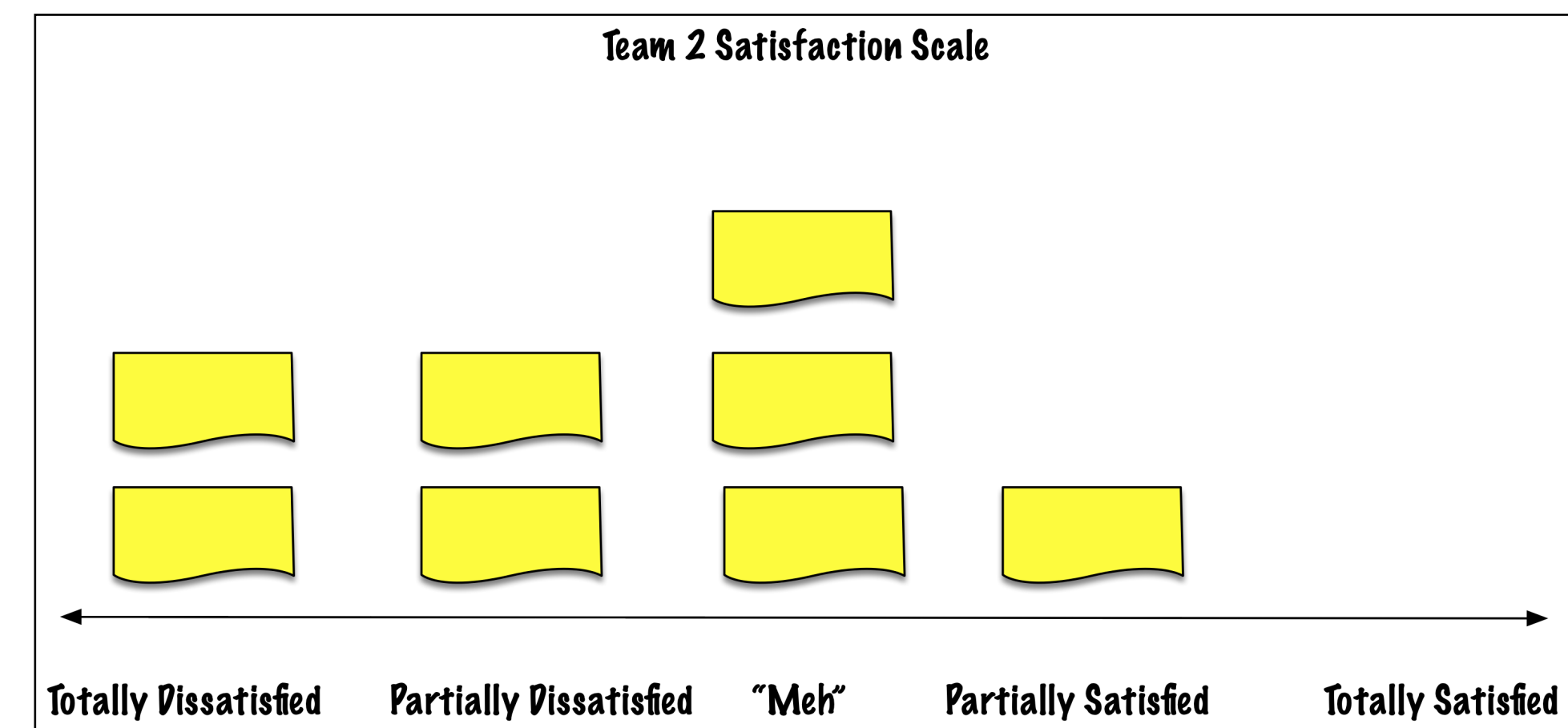
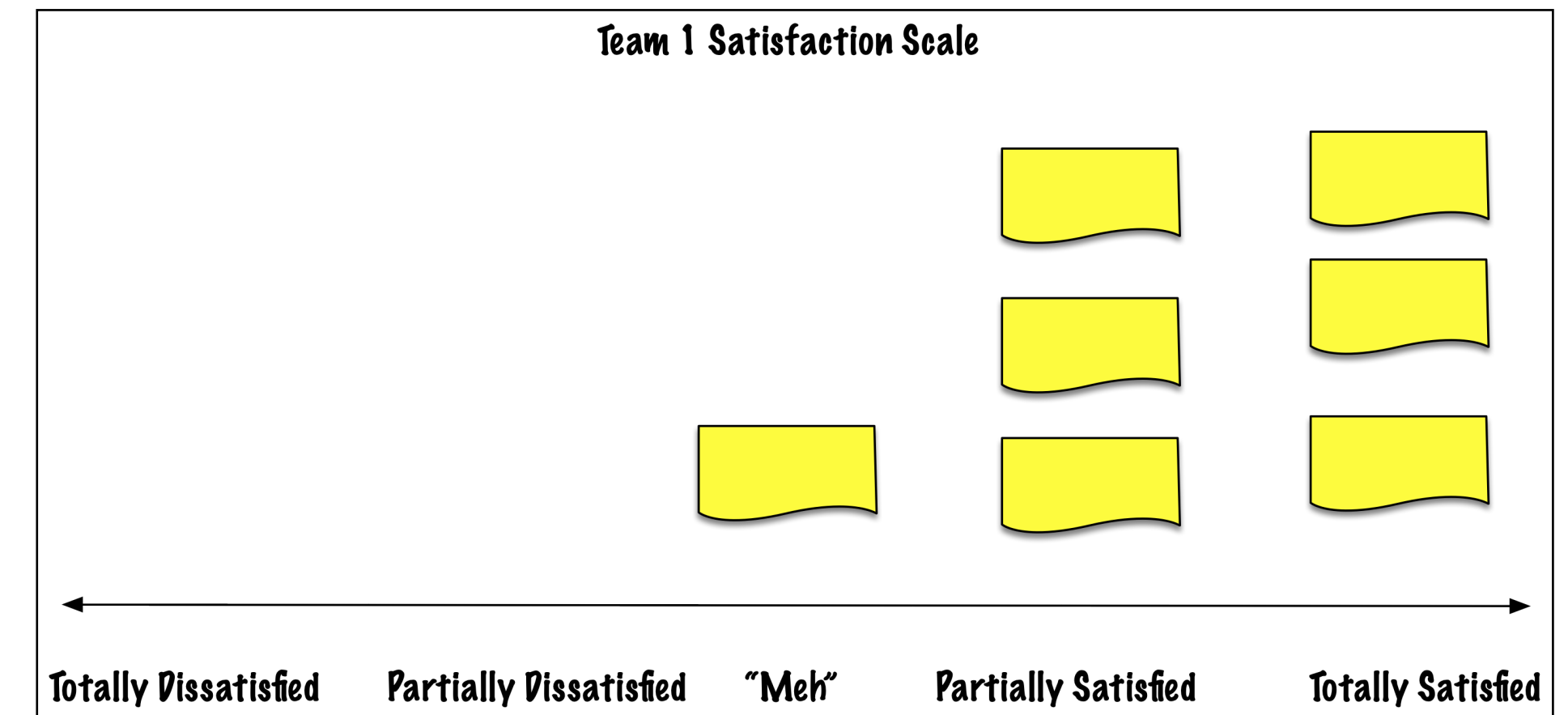
- More data about wait times:
 - When the work waits, people are doing something else. What is that?
 - What is the relative ranking of *this* story and *this* project? (Is there more important work?)
- Prioritization is NOT the same as ranking. Rank all *work*. Stop ranking *people* because they are capable of much more value.

Rank Order



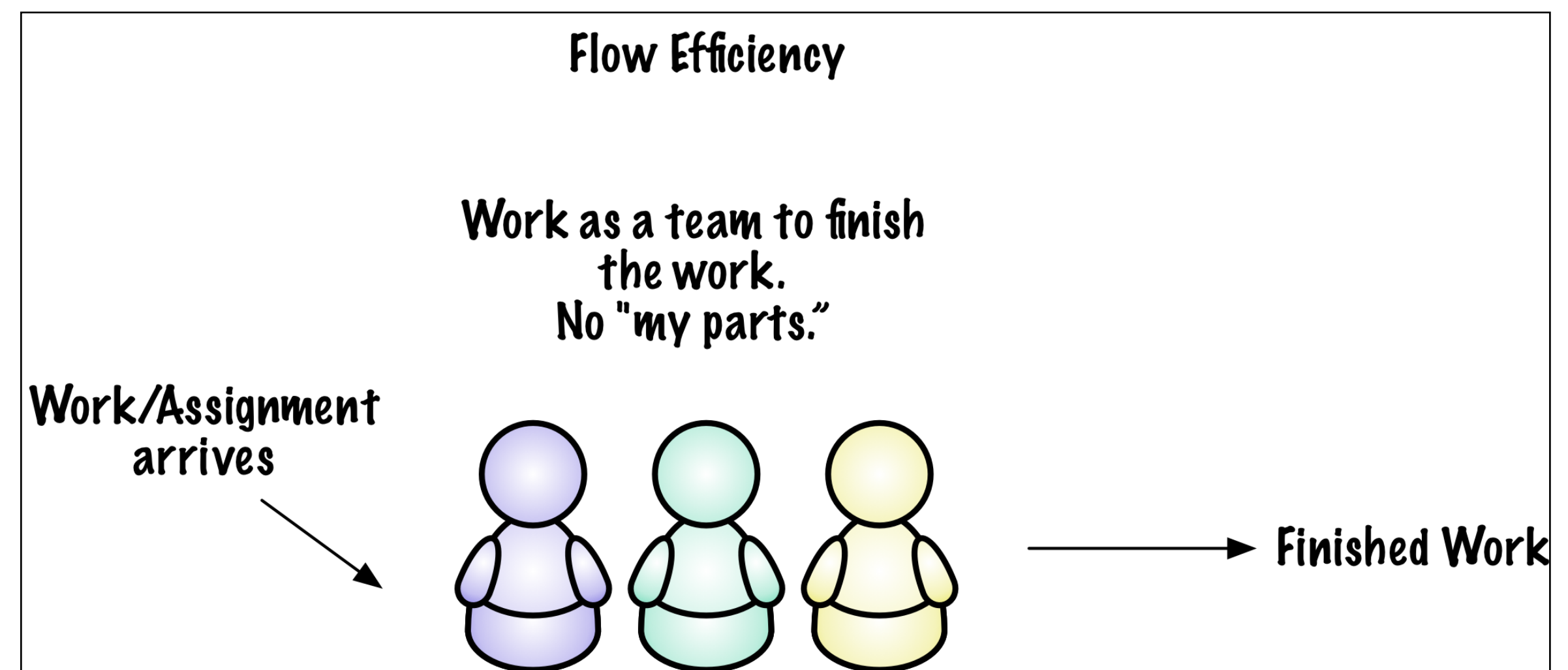
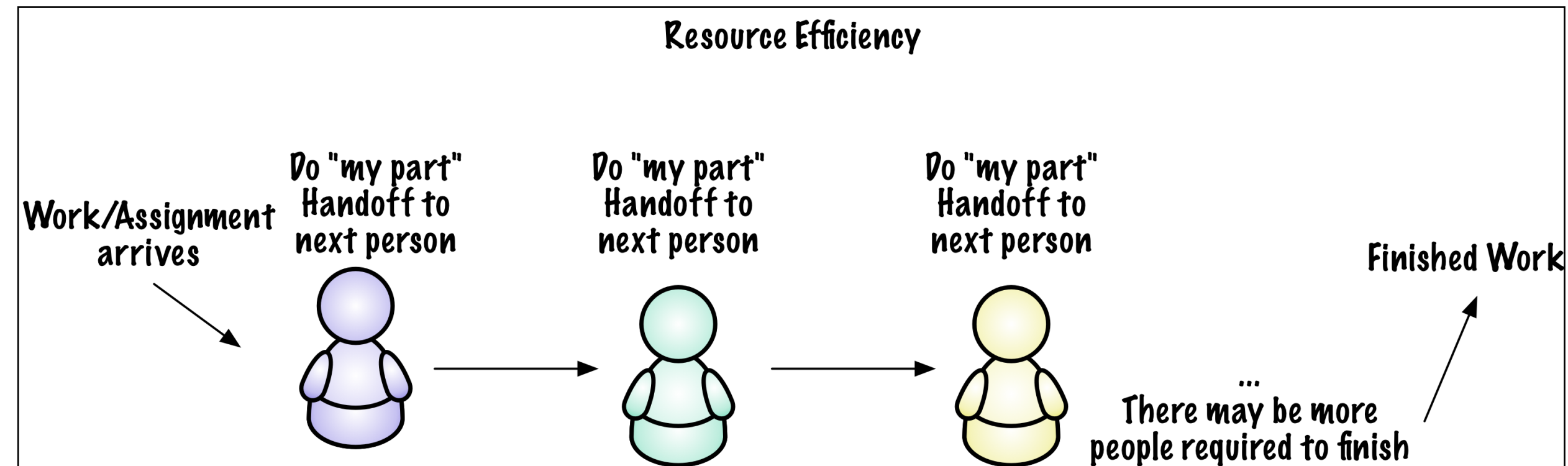
Ask About Subjective Data

- Subjective data for wait times:
 - Work satisfaction throughout the work (daily, as a project trend, etc.)



Resource Efficiency vs Flow Efficiency

- Resource efficiency thinking:
Individual actions or deliverables:
outputs
- Flow efficiency thinking:
Focus on collaborative *outcomes* that
serve an overarching goal
- Rewards drive your organizational
culture



Very few teams have just three
people...

Add More People?

- Brooks' Law meets Little's Law
 - When people work as experts, Brooks' Law holds (adding more people to a late project makes it later)
 - But when a team collaborates, WIP matters more than the number of people
- That's why collaboration *time* matters more than physical location for distributed teams

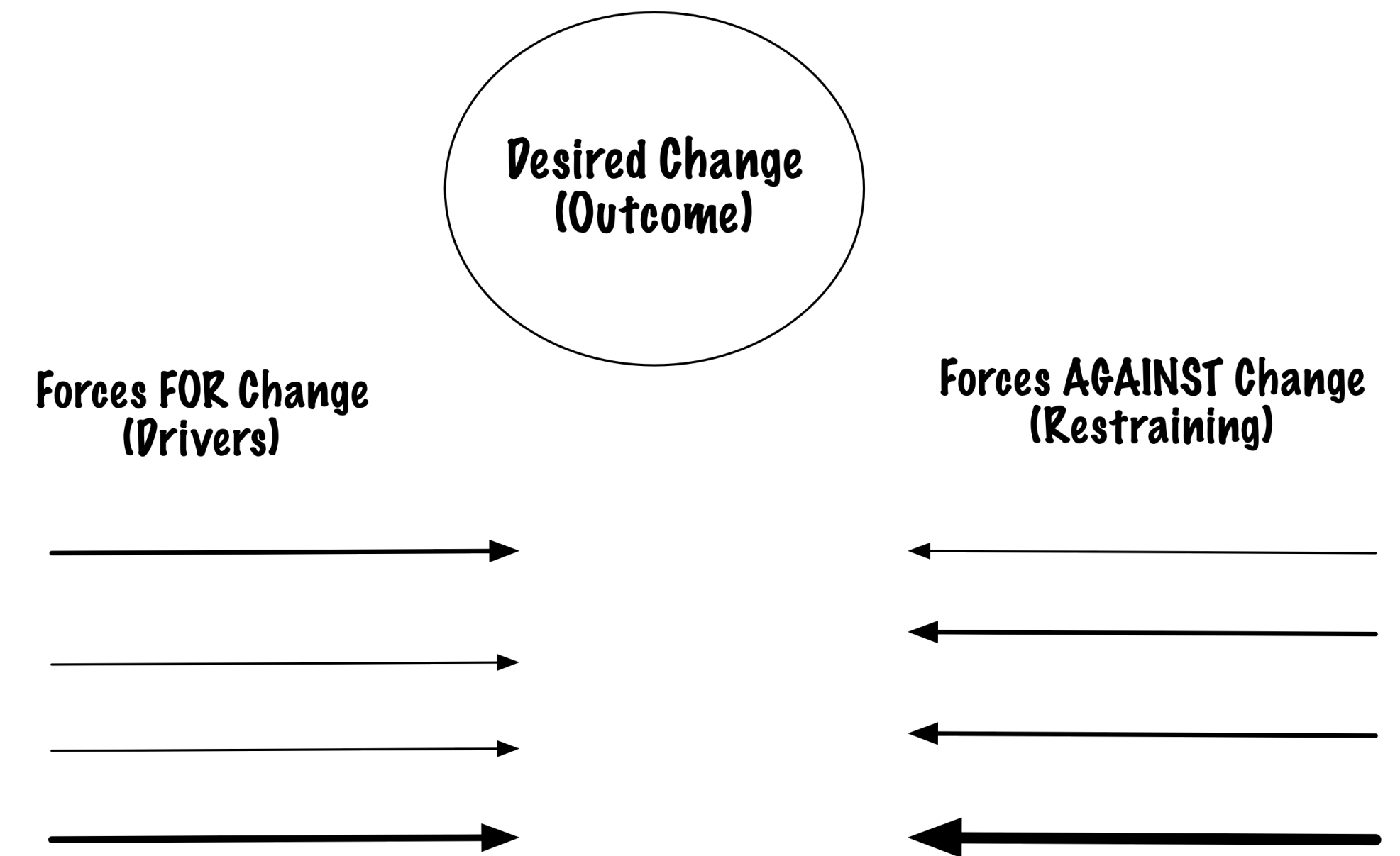
Little's Law:

$$\text{Work in Progress (WIP)} = \text{Cycle Time} * \text{Throughput}$$

Choices to Increase Agility

Solve Problems as a Team

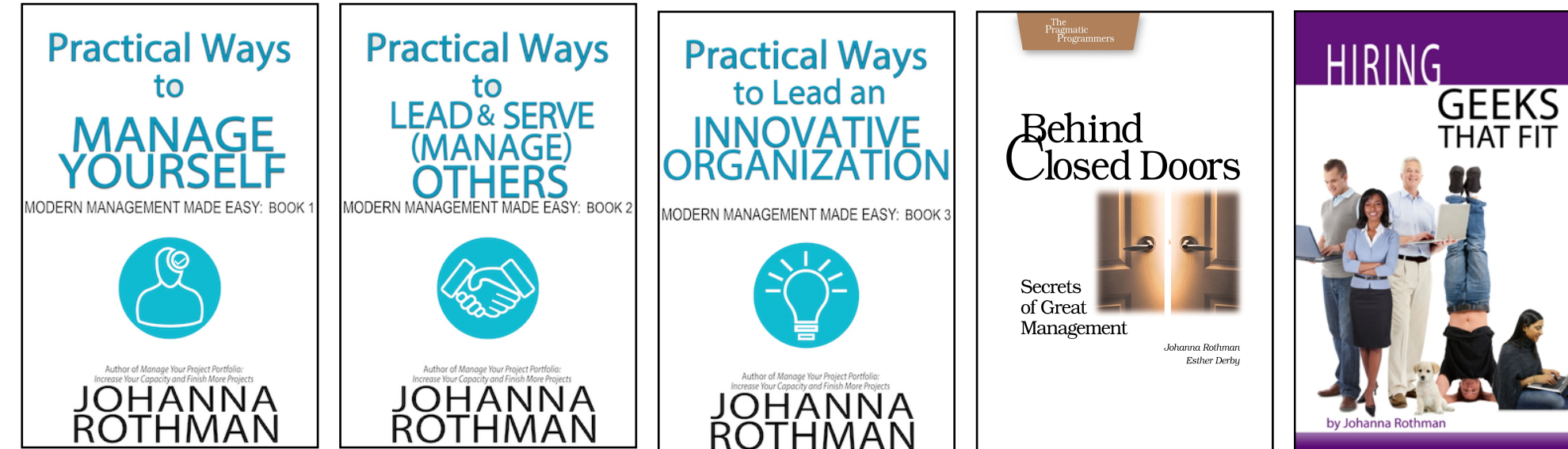
- Where can you, as a team, choose to collaborate?
- What will ease your collaboration or make it more difficult?
- Consider a Force Field Analysis
- Collaboration can be expert-based as in swarming



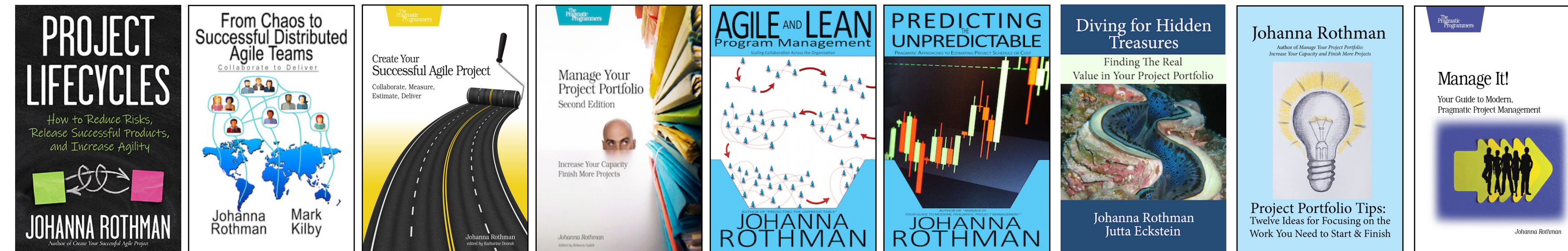
Collaboration in some way will ease your “too
long” problems
&
It might be more fun!

All My Books (Organized)

Management



Product Development



Personal Development



Let's Stay in Touch

- Pragmatic Manager: www.jrothman.com/pragmaticmanager
- Please link with me on LinkedIn: <https://www.linkedin.com/in/johannarothman/>
- Start here for the blog posts: <https://www.jrothman.com/mpd/2026/03/how-to-use-value-stream-maps-to-reinforce-agility-effectiveness-part-1/>

