

# Feature Teams (vs. Component Teams)

*Exposing Local Optimization*

***presented @  
UnitedHealth Care  
Lunch & Learn***

*by Gene Gendel*

# Local Optimization

# Local Optimization

*(“This Is Not My Job”)*



Sourced from: [https://static.timesofisrael.com/www/uploads/2019/11/000\\_1MB1Q8-640x400.jpg](https://static.timesofisrael.com/www/uploads/2019/11/000_1MB1Q8-640x400.jpg)

# Local Optimization

...whereas, **Global optimization** refers to finding the optimal value of a given function among all possible solution...

... **Local optimization** finds the optimal value within the neighboring set of candidate solution...

<https://www.lqi-global.com/dictionary/from-optimization-to-clustering/45858>

[Pentagon Wars – Bradley Fighting Vehicle Evolution](#)



# Local Optimization

*"Everyone is busy and working so hard. Yet, the system is delivering slow and Users are not happy"*

How could that be?



Sourced from: <https://www.youtube.com/watch?v=5unMIXg6WL4>

# Local Optimization

## Frequently Heard Justifications:

- Efficient
- Productive
- Best
- Good
- Optimized
- Ideal
- Cost-Effective



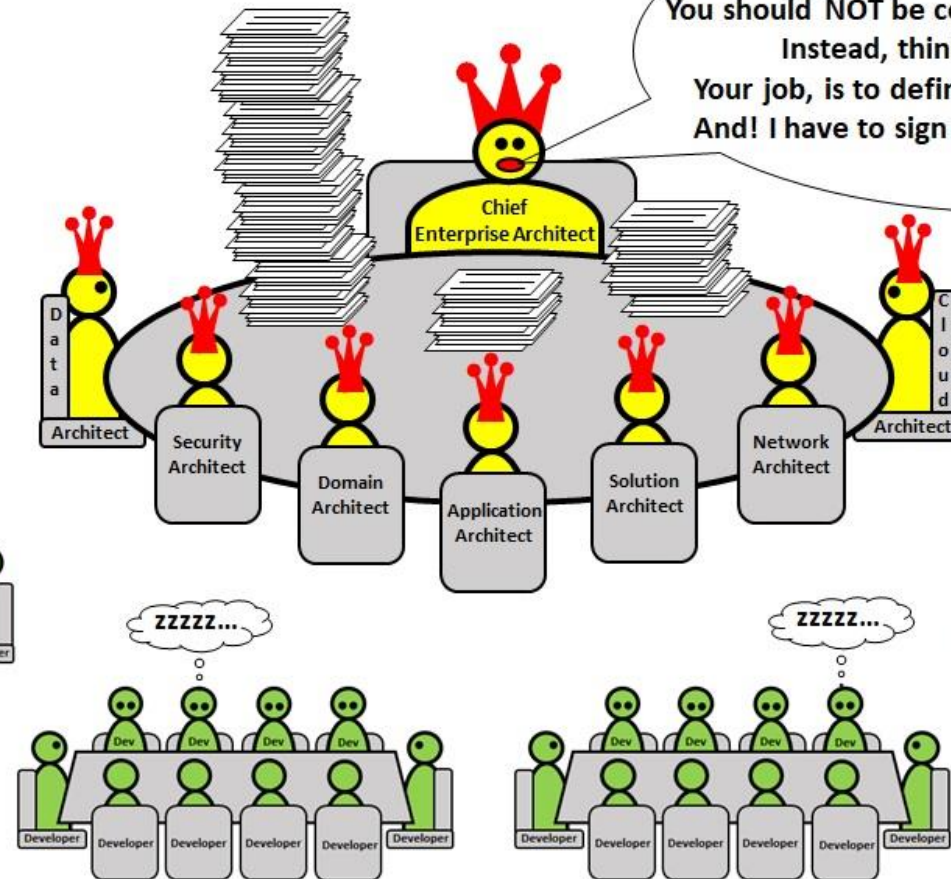
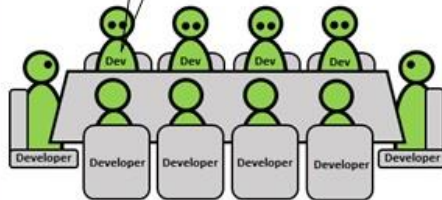


# Local Optimization

## “Power Point Architects” In Action



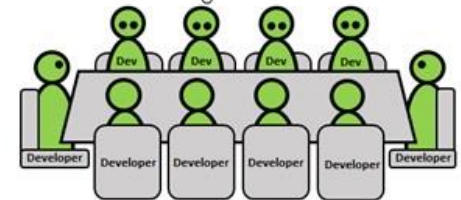
Guys! We are Master-Programmers, with many decades of experience. How come, we are not allowed to define architecture???



**Remember!!!**

You are the ELITE of IT!  
You should NOT be coding! Leave it to developers!  
Instead, think about architecture.  
Your job, is to define best practices for others.  
And! I have to sign off on all of your decisions.

Oh God... Another mandate from the "Ivory Tower"???



Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>

# Local Optimization

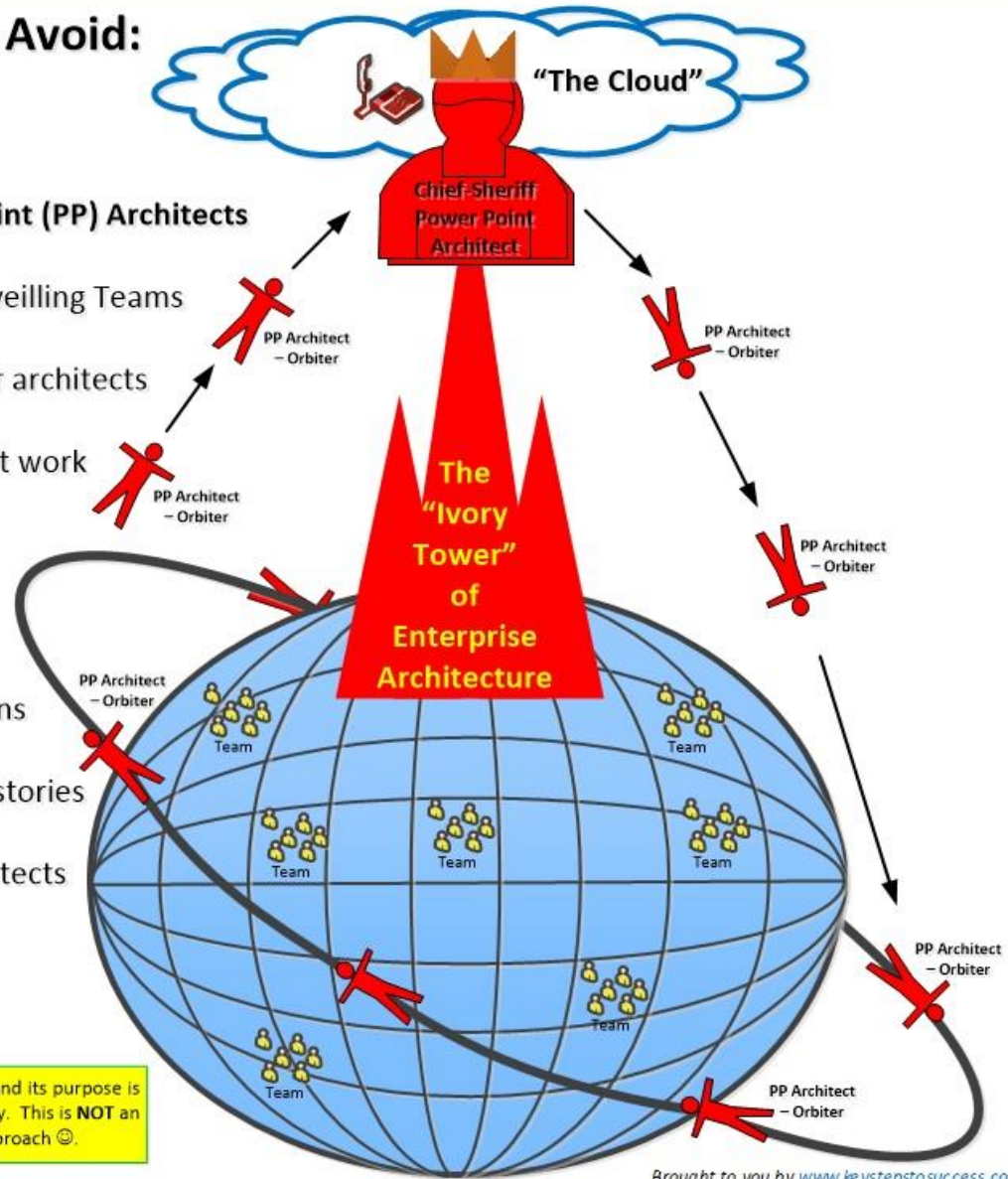
## Typical Problems to Avoid:

- Building architecture 'towers'/silos
- Reliance on **Chiefs-Sheriffs Power Point (PP) Architects**
- Squads of **PP Architects-Orbiters**, surveilling Teams
- Separate reporting structures, just for architects
- Architects, not doing any development work
- Local Optimization in architecture
- Architecture "away" from Business
- "One-size-fits-all" architecture solutions
- Architecture work streams, backlogs, stories
- Reliance on expensive vendors - architects



Avoid This:

**Attention:** This graphic is a cartoon and its purpose is to expose a dysfunction through irony. This is **NOT** an invitation to experiment with this approach 😊.

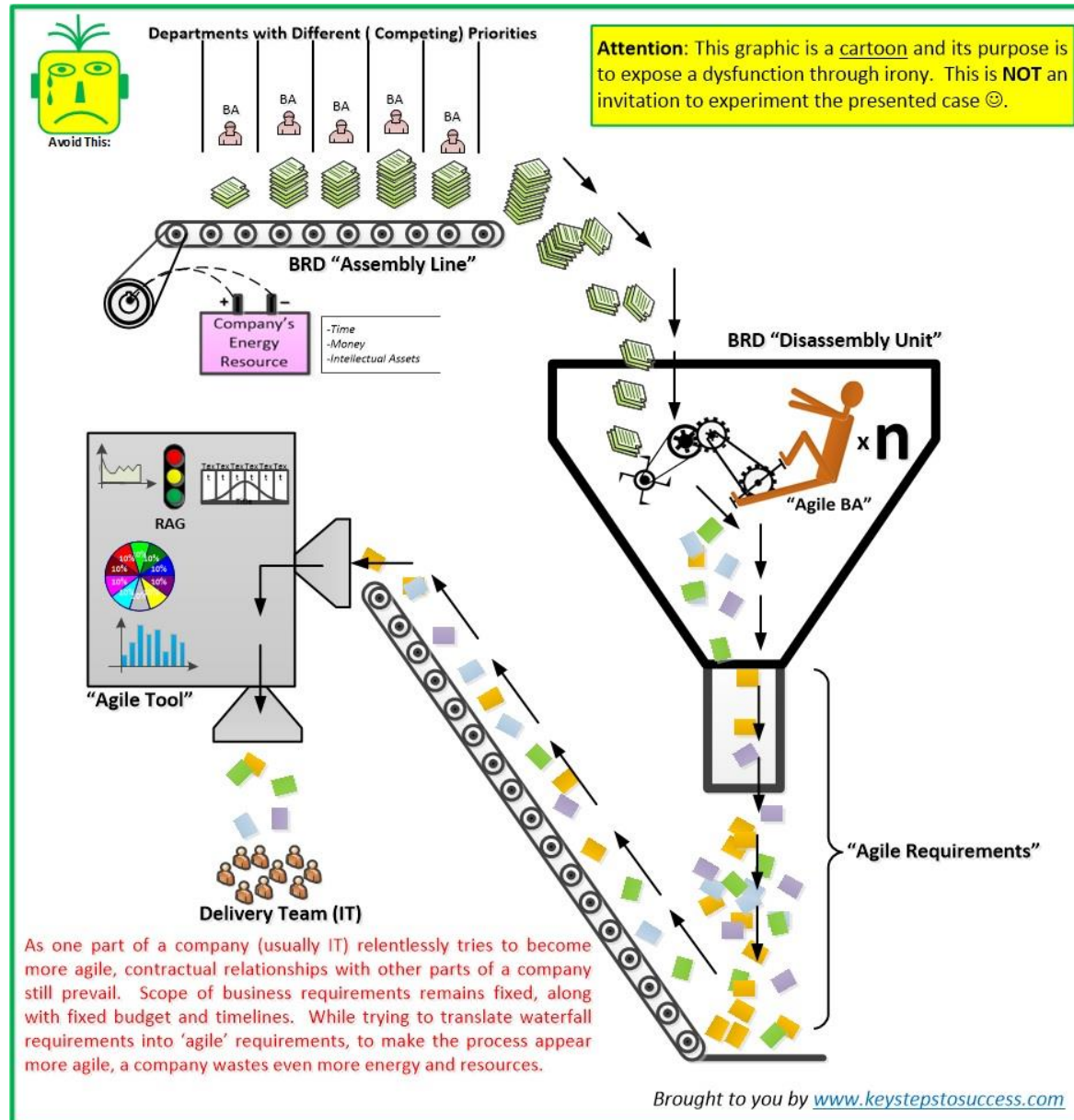


Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>

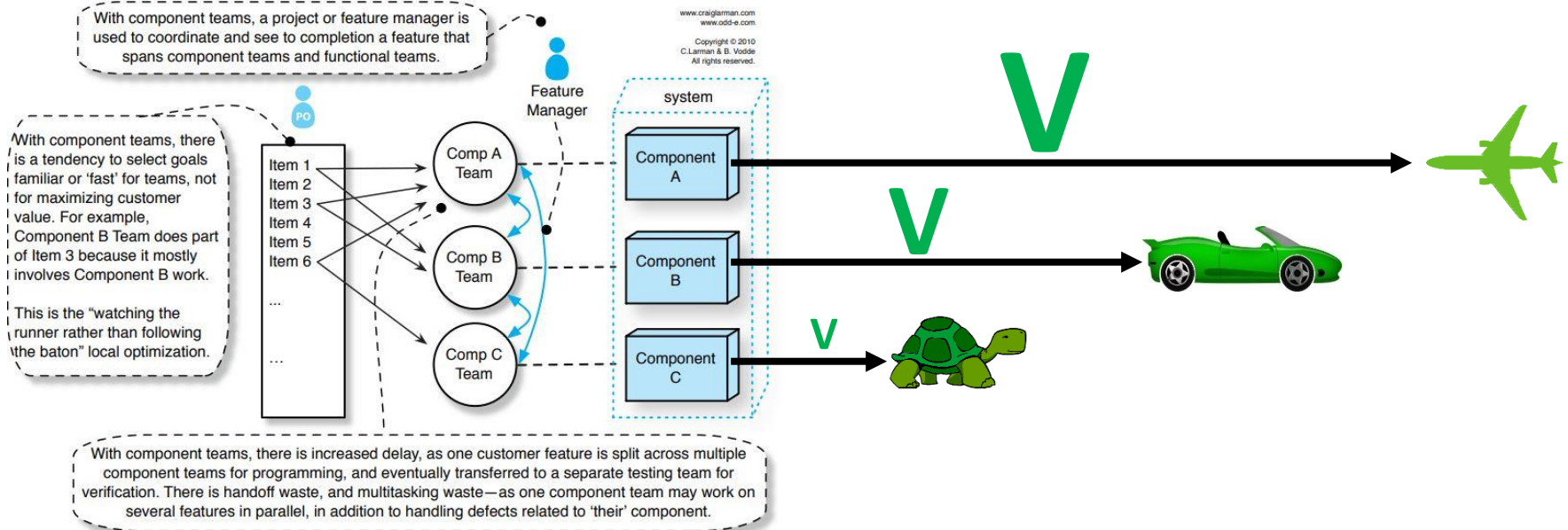


# Local Optimization



# LeSS Guides: *Organize by Customer Value*

## Component Teams



Sourced from: <https://less.works/resources/graphics/index.html>

### Big Mistake # 1

$$V > V > v = \text{Thinking Face}$$

### Big Mistake # 2


$$V + V + v = \text{Thinking Face}$$

***Can these velocities be used to reliably estimate volume and complexity of delivered features?***

# LeSS Guides: *Organize by Customer Value*

*Synonyms for “Component Team”*

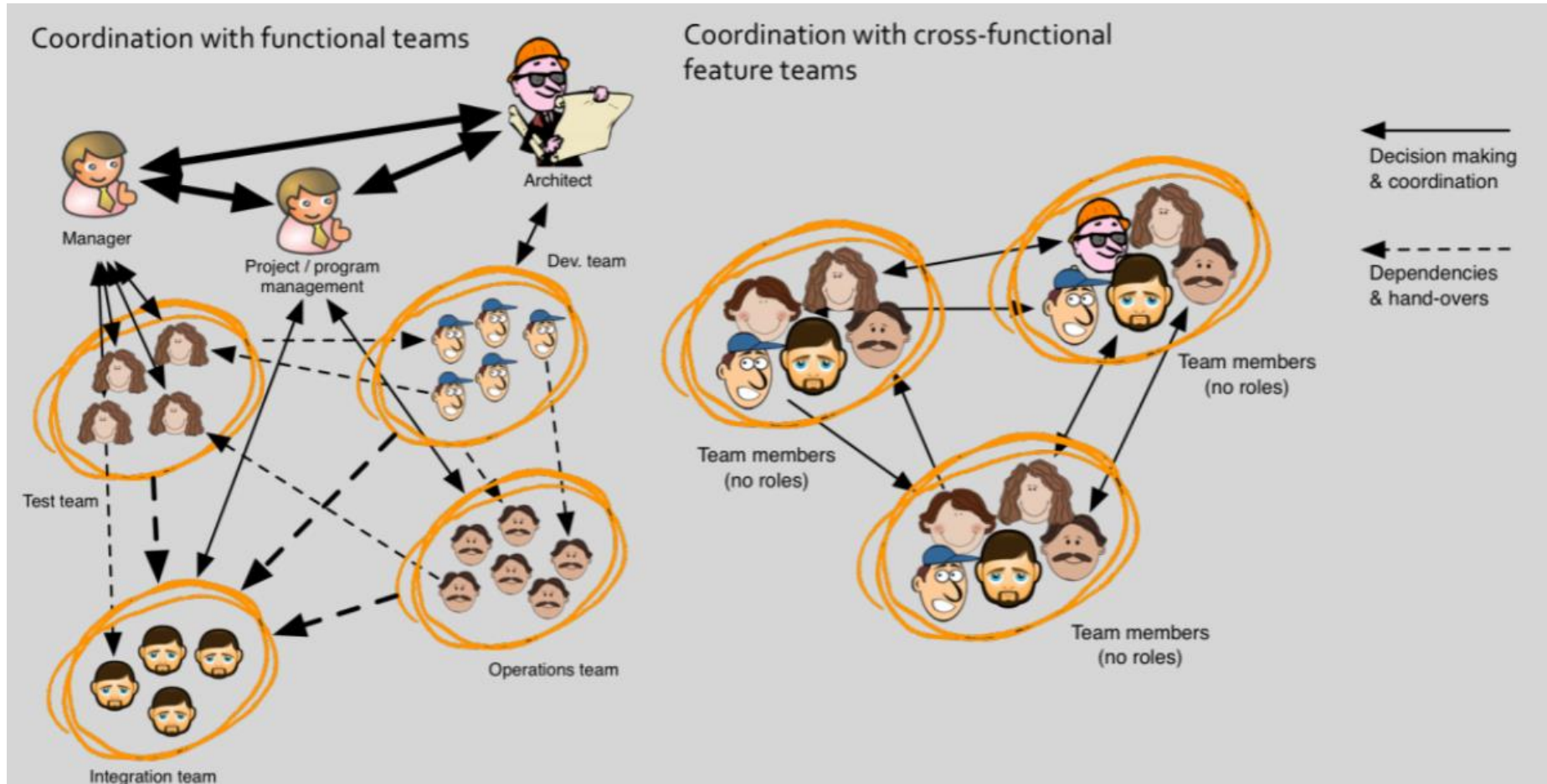
Local Optimization



platform team  
core or shared services team  
micro-service team  
application team  
subsystem team  
library team  
service team  
API team  
front-end (or back-end) team  
DB team  
module team  
framework team  
DDD bounded-context team

Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>

# LeSS Guides: *Organize by Customer Value*

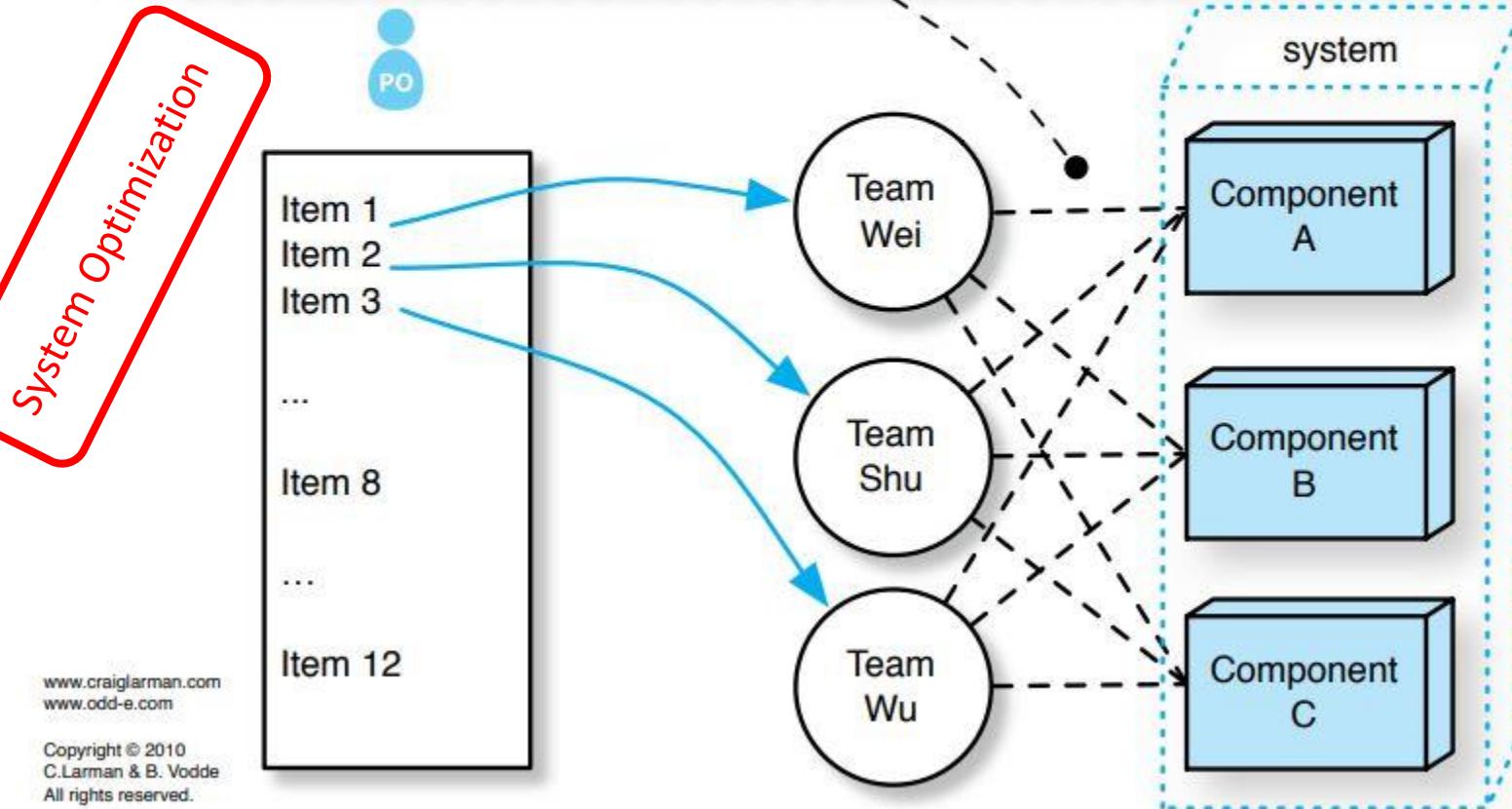


Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Organize by Customer Value*

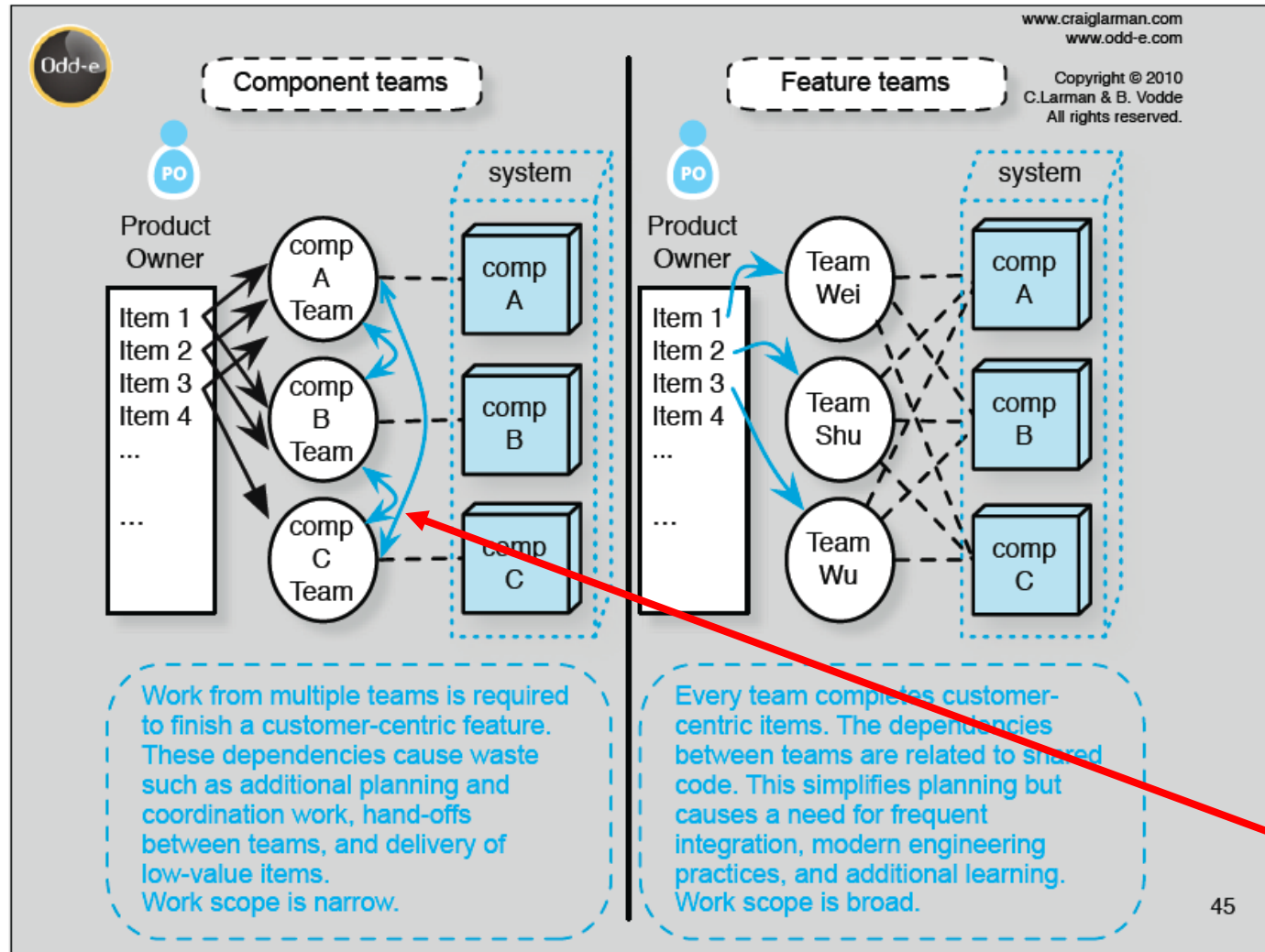
With feature teams, teams can always work on the highest-value features, there is less delay for delivering value, and coordination issues shift toward the shared code rather than coordination through upfront planning, delayed work, and handoff. In the 1960s and 70s this code coordination was awkward due to weak tools and practices. Modern open-source tools and practices such as TDD and continuous integration make this coordination relatively simple.



Sourced from: <https://less.works/resources/graphics/index.html>



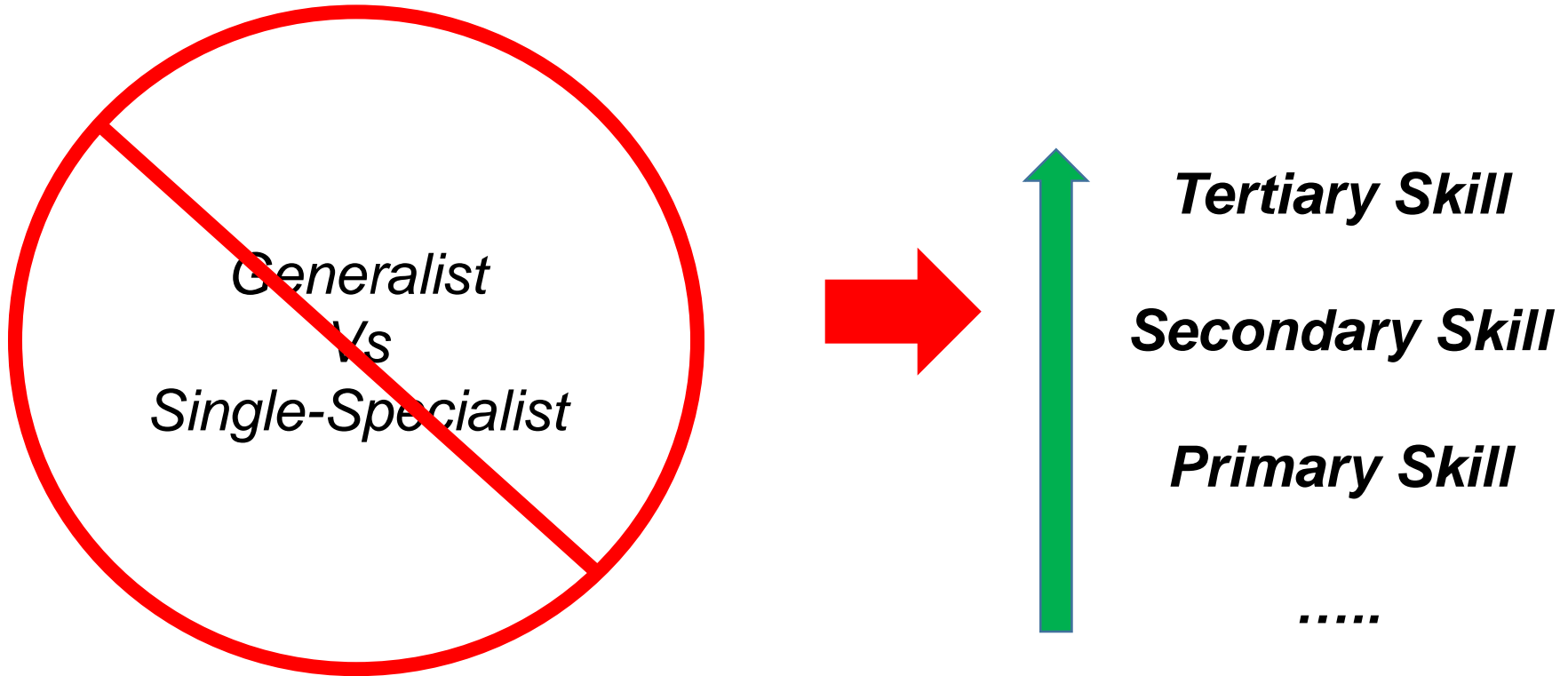
# LeSS Guides: *Organize by Customer Value*



**Coordination Costs**  
\$\$\$\$\$\$\$\$\$\$\$\$

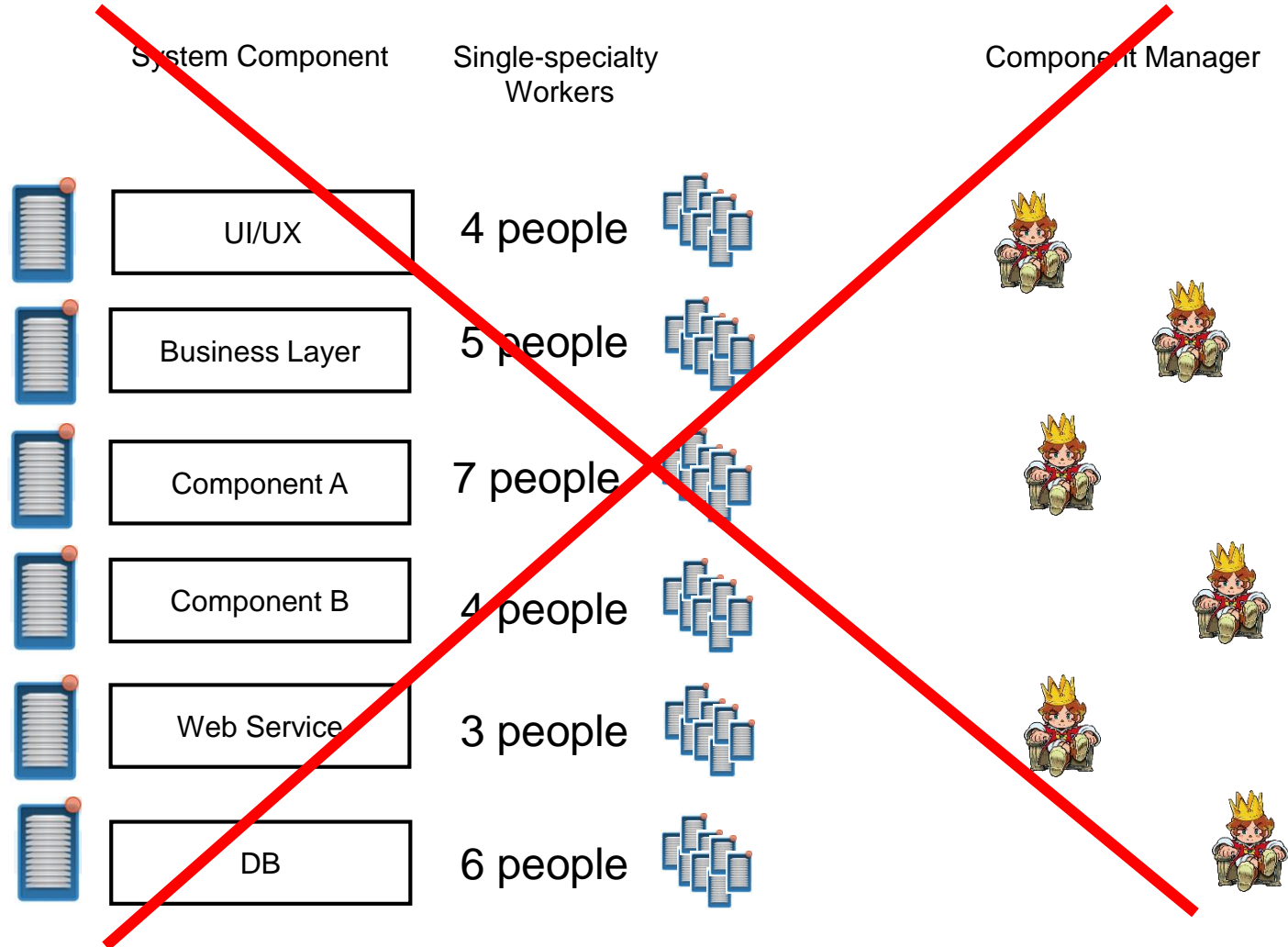
# LeSS Guides: *Organize by Customer Value*

## *Avoiding False Dichotomies*

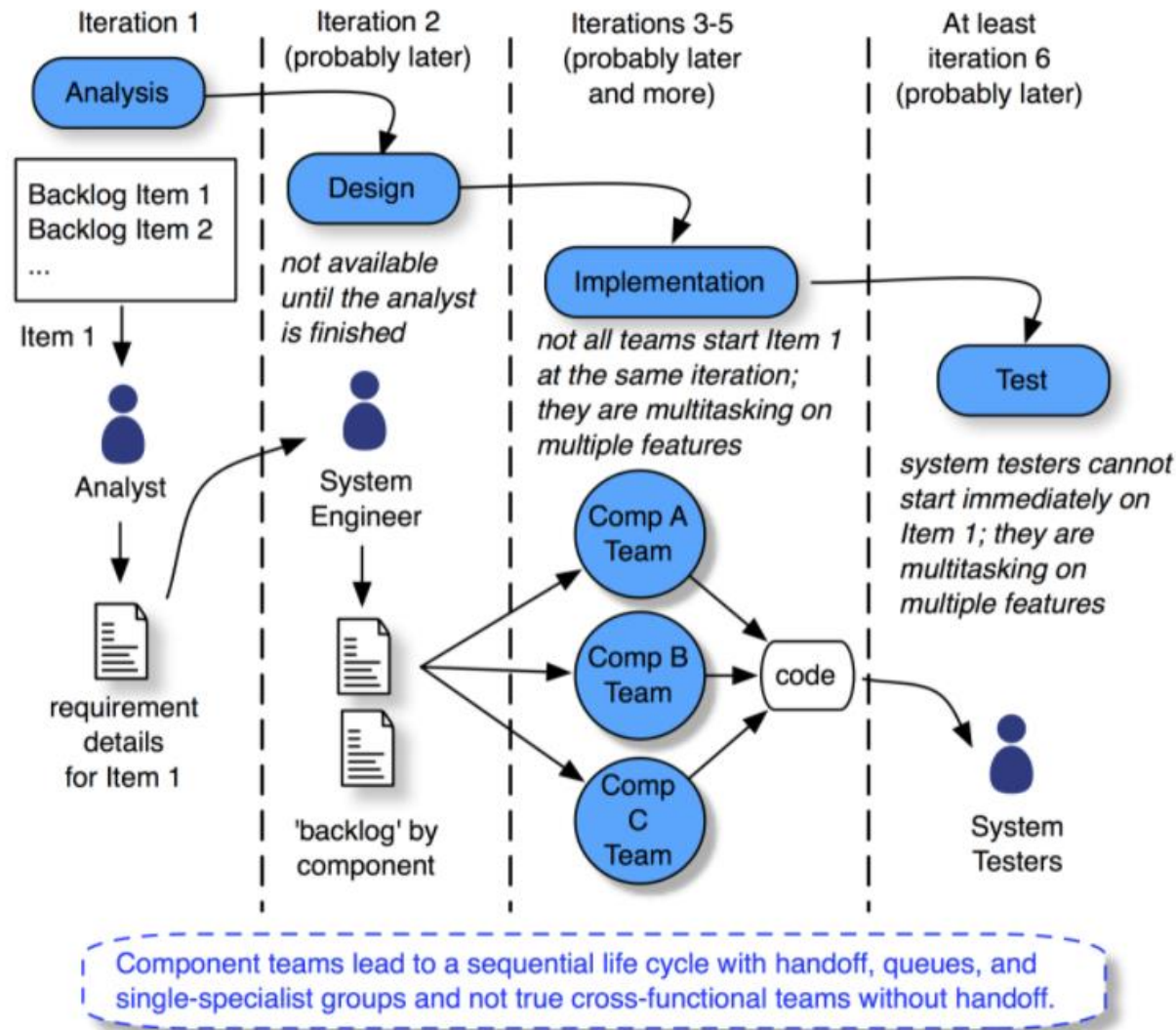


# LeSS Guides: *Organize by Customer Value*

## *"Back story" of Component Teams*

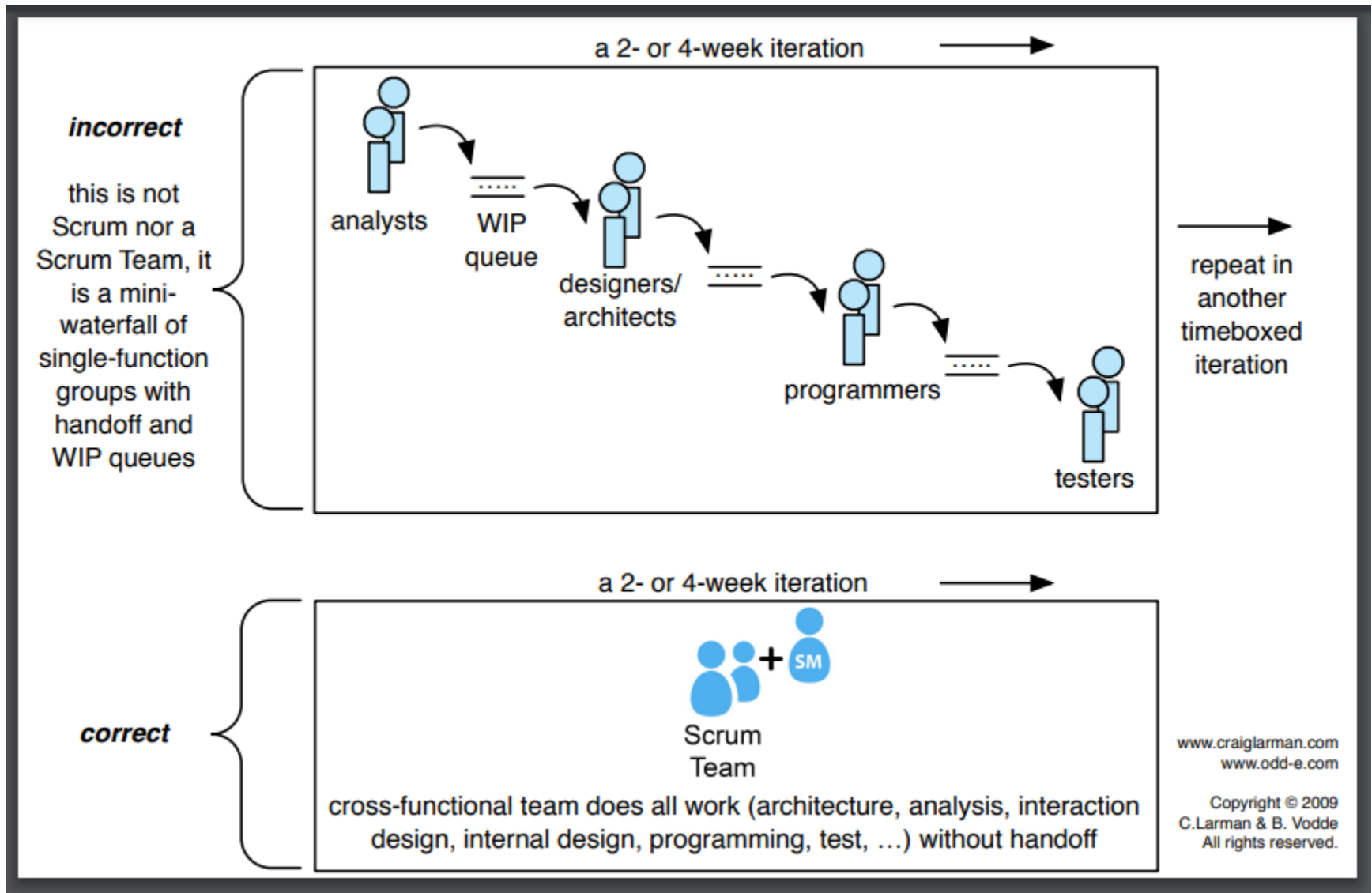


# LeSS Guides: *Organize by Customer Value*



Sourced from <https://less.works/less/structure/feature-teams>

# LeSS Guides: *Organize by Customer Value*



Sourced from <https://less.works/scaling-book-images/scaling-agile-lean-development-thinking-tools/chapter-8-single-function-groups-en.pdf>



# LeSS Guides: *Organize by Customer Value*

## Seeing (and Hearing) Local Optimization

“Everyone is doing their best yet overall systems throughput is degrading. How can that be?” This is the paradox of **local optimization** —when a person or departmental decision maker optimizes for the local view or self-interest. The party making the decision frequently *believes they are making the best decision*, but because ‘best’ is a local optimization, in fact it sub-optimizes overall system throughput. This is a result of “silo mentality,” misunderstanding, fear, limited information, delayed feedback, ignorance, careerism, avarice, and other common *organizational learning disorders*.

Team Structures

Org. Structures

Documentation

Definition of Done

Backlogs

Role Definitions

Product Design

Goals & Metrics

# Q & A