Troy Magennis (@t_magennis)

Entangled: Solving the Hairy Problem of Team Dependencies

Entangled: Solving the Hairy Problem of Team Dependencies

Troy Magennis

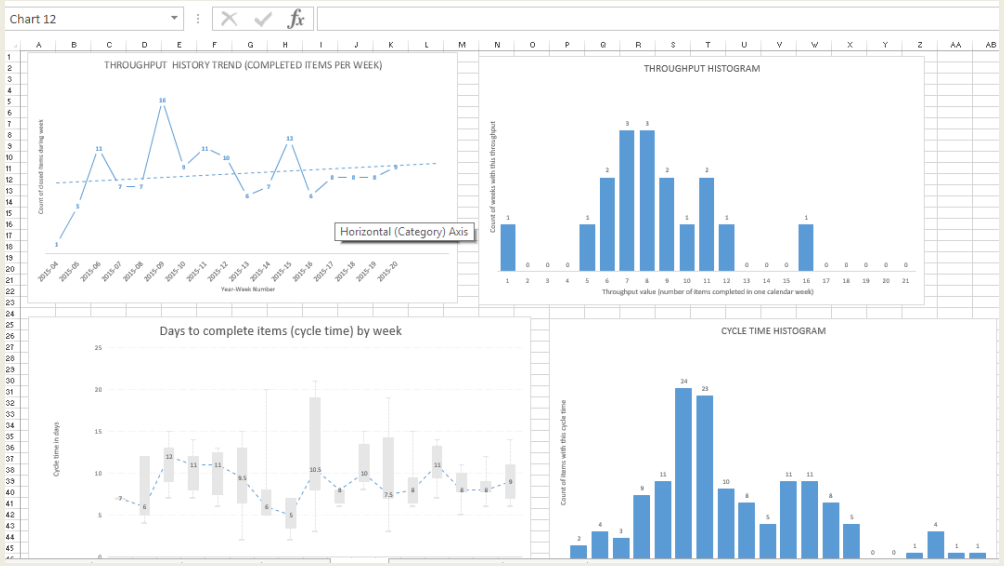
@t_magennis

Troy.Magennis@focusedobjective.com

Slides, Forecasting Spreadsheets, Resources

Bit.Ly/SimResources

(case sensitive)



Dependency - progress of one action relies upon the timely output of a previous action, or the presence of some specific thing.

From: Strode, D.E. and Huff, S.L. A taxonomy of dependencies in agile software development. 23rd Australasian

A Taxonomy of Dependencies in Agile Software Development

Diane E. Strode
Sid L. Huff
School of Information Management
Victoria University of Wellington
Wellington, New Zealand
Email: diane.strode@alumni.unimelb.edu.au
Email: sid.huff@vuw.ac.nz

Abstract

Dependencies in a software project can contribute to unsatisfactory progress if they constrain or block the flow of work. Various studies highlight the importance of dependencies in the organisation of work; however dependencies in agile software development projects have not previously been a research focus. Drawing on three case studies of agile software projects, and the IS literature, this paper develops an initial taxonomy of agile software project dependencies. Three distinct categories of dependency are found: task, resource, and knowledge dependencies. This paper contributes to theory by providing a taxonomy of dependency types occurring in the area of agile software development. Practitioners can use this taxonomy as sensitising device to ensure they consider dependencies they might face that could hinder their projects, enabling them to take appropriate and timely mitigating action.

Keywords

Agile software development, Dependency analysis, Dependency Taxonomy, Software project dependencies.

https://dro.deakin.edu.au/eserv/DU:30049080/strode-taxonomyofdependencies-2012.pdf

Taxonomy of Agile Software Project Dependencies

Knowledge Dependency

Requirement | Expertise | Task Allocation | Historical

Requirement - a situation where domain knowledge or a requirement is not known and must be located or identified and this

Task Dependency

Activity | Business Process

nother task can proceed and this affects project progress. There

Activity - a situation where an activity cannot proceed until another activity is complete and this affects project progress

Resource Dependency

Entity | Technical

ects progress, such as when one software component must

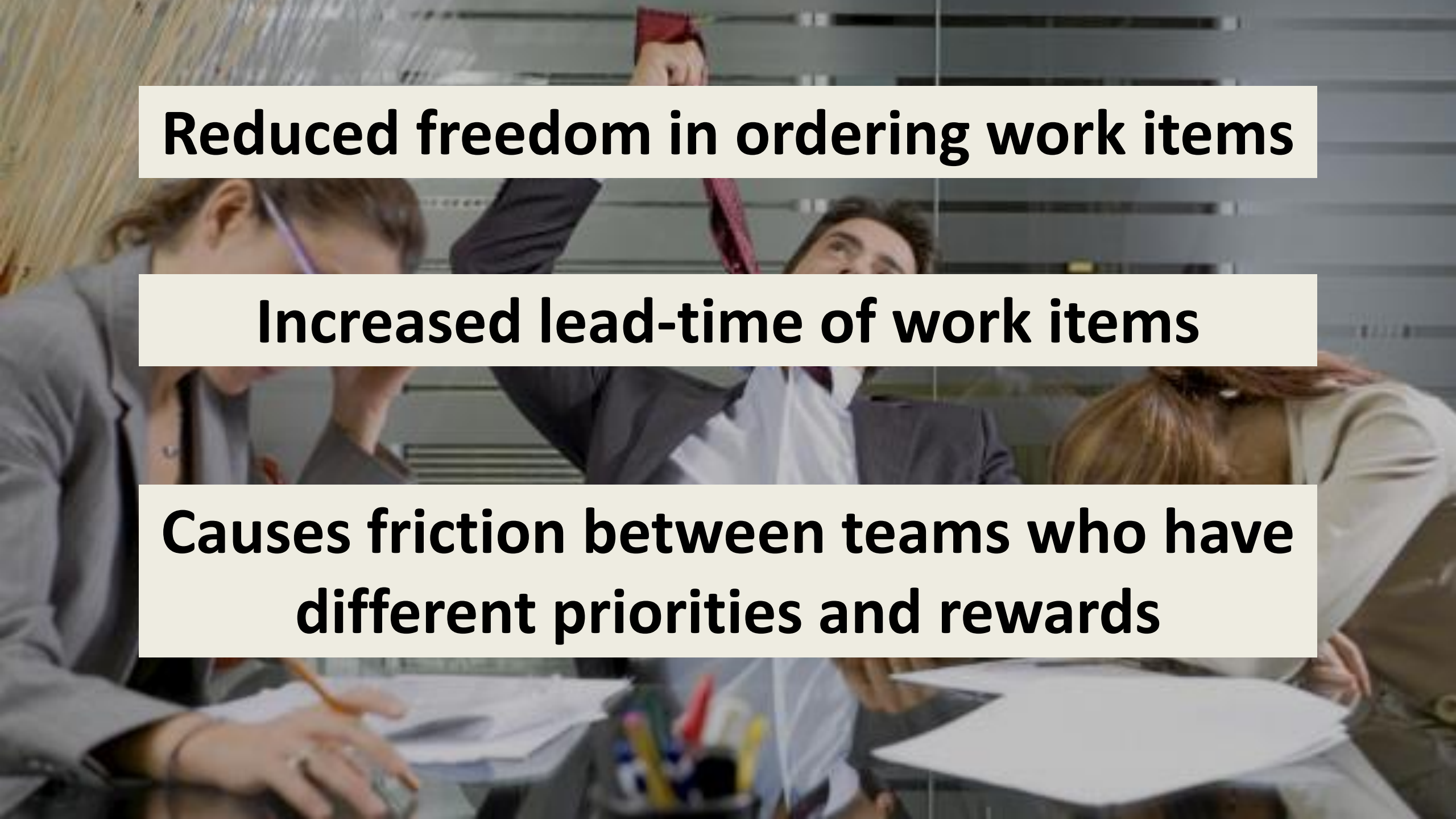
ence affects project progress

Figure 1: A taxonomy of dependencies in agile software development projects

https://dro.deakin.edu.au/eserv/DU:30049080/strode-taxonomyofdependencies-2012.pdf

Dependency - progress of one action relies upon the timely output of a previous action, or the presence of some specific thing.

From: Strode, D.E. and Huff, S.L. A taxonomy of dependencies in agile software development. 23rd Australasian



Reduced freedom in ordering work items

Increased lead-time of work items

Causes friction between teams who have different priorities and rewards

REDUCED FREEDOM IN ORDERING WORK ITEMS

The Effects of Team Backlog Dependencies on Agile Multiteam Systems: A Graph Theoretical Approach

Alexander Scheerer
SAP SE &
University Mannheim
scheerer@uni-mannheim.de

Saskia Bick
SAP SE &
University Mannheim
bick@uni-mannheim.de

Tobias Hildenbrand
SAP SE
tobias.hildenbrand@sap.com

Armin Heinzl
University Mannheim
heinzl@uni-mannheim.de

Abstract

In agile software development, the prioritization of backlog items is the mission-critical responsibility of the product owners in order to maximize the customer value created by development teams. However, in the reality of large-scale development, the degree of freedom for such a prioritization is substantially restricted by various types of interdependencies between backlog items. In this work, we show, using a graph theoretical approach, the relation between the degree of freedom for prioritization and the occurrence of dependencies. To the best of our knowledge, the breadth and depth of such consequences has never been modeled or investigated up until now. Based on our results, we derive implications for real-world large-scale software development in agile environments.

dependencies to allow for effective multiteam system coordination and efficient intra-team work [45]. Yet, in real-world examples of large-scale agile development environments, product owners are often complaining about being responsible for prioritizing requirements and creating value for customers, while the stage is already set due to given dependencies. As described in the Scaled Agile Framework (SAFe) by Leffingwell [26], cross-team features in the shared program backlog are split into single-team backlog items, i.e. user stories, with tentative allocations to individual sprints, which limits the product owners and their teams in their "free choice" of backlog item prioritization.

According to the agile development paradigm, maximizing customer value and thus ensuring economic product viability for the company is a key principle [38]. Nevertheless, products are rarely built from

http://conferences.computer.org/hicss/2015/papers/7367f124.pdf

four dependencies (see Figure 5). The data shows strikingly, an increased amount of dependencies dramatically lowers the DoF. This confirms our initial expectation in chapter 2.2, that predetermined dependencies have a significant impact on the prioritization ability of the team product owner. In fact, the product owner is drastically limited in freely assigning priorities to the backlog items and subsequently ordering them accordingly. One dependency between two backlog items, for example, will already reduce his freedom of choice in how to prioritize the backlog by 50%.

http://conferences.computer.org/hicss/2015/papers/7367f124.pdf

Feature A	Feature B	A before B
1	2	Yes
2	1	No

Feature start order

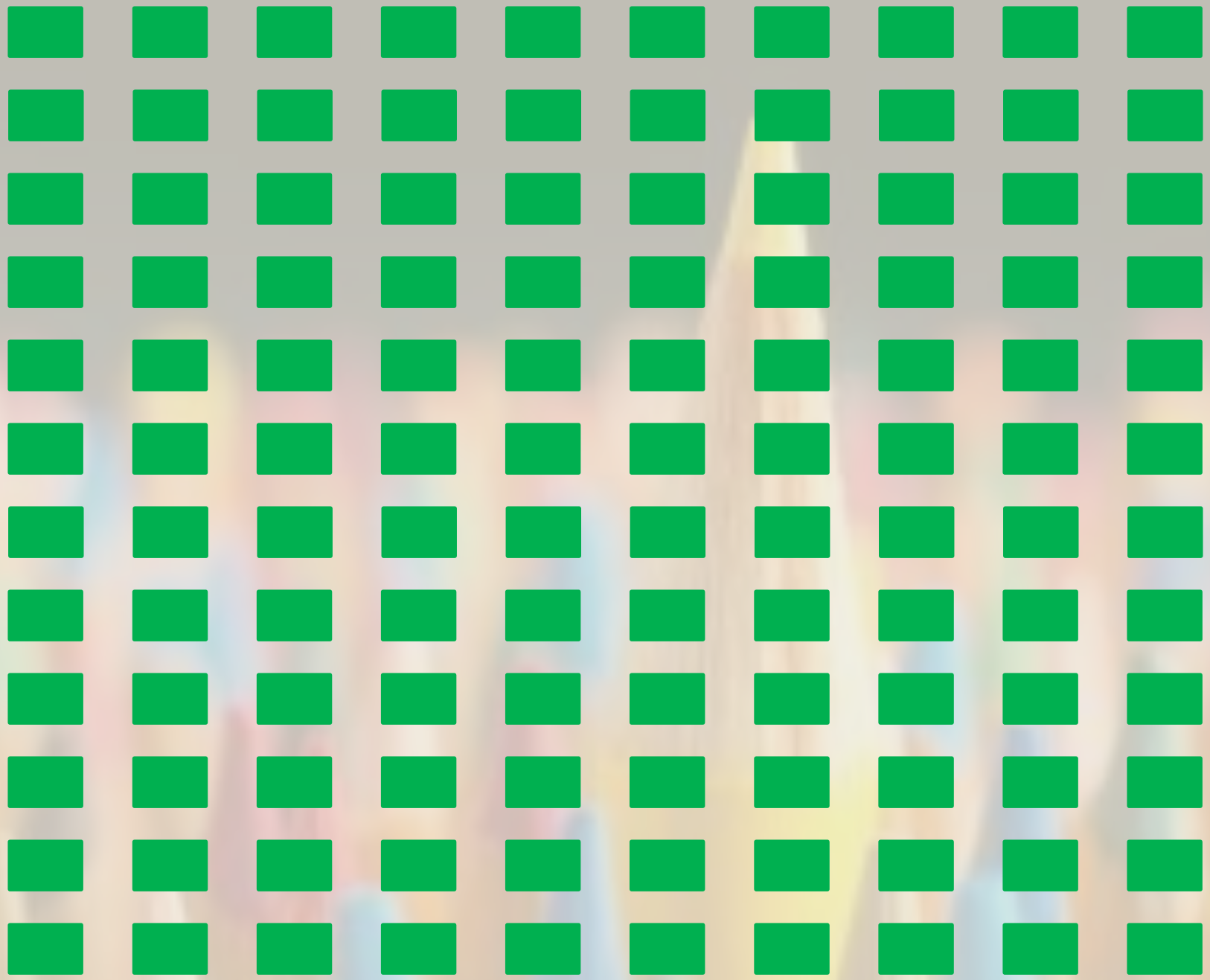
One dependency cuts allowed options in half

Two dependencies cuts allowed options to 1/6th

Feature A	Feature B	Feature C	A < B	A < B, B < C
1	2	3	Yes	Yes
1	3	2	Yes	No
2	1	3	No	No
2	3	1	No	No
3	1	2	Yes	No
3	2	1	No	No

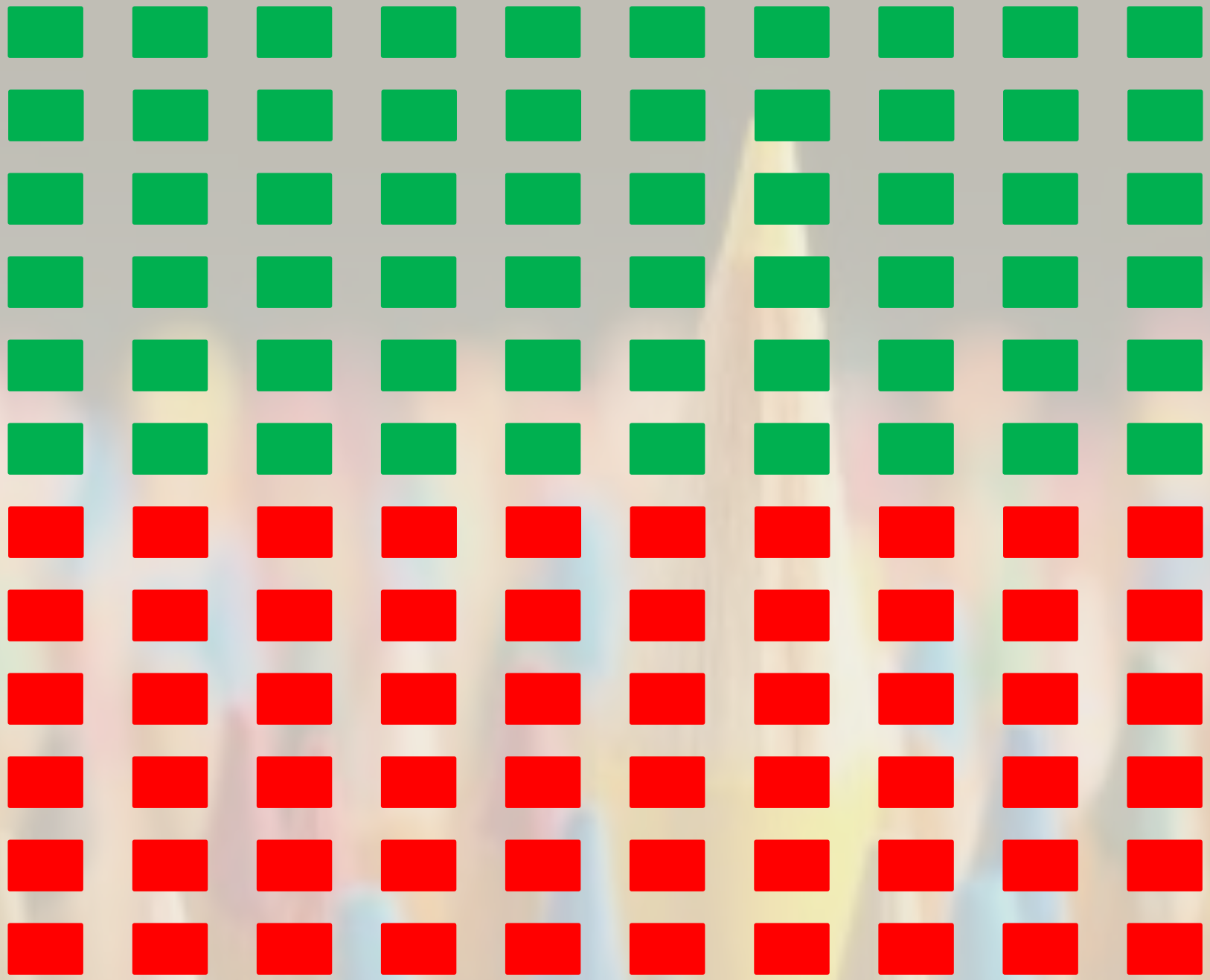
Number of Dependencies	Valid Ordering Options	# valid options for 4 features	# valid options for 5 features	# valid options for 6 features
0	100%	24	120	720
1	50%	12	60	360
2	16.667%	4	20	120
3	4.167%	1	5	30
4	0.833%	-	1	6
5	0.278%	-	-	1

0



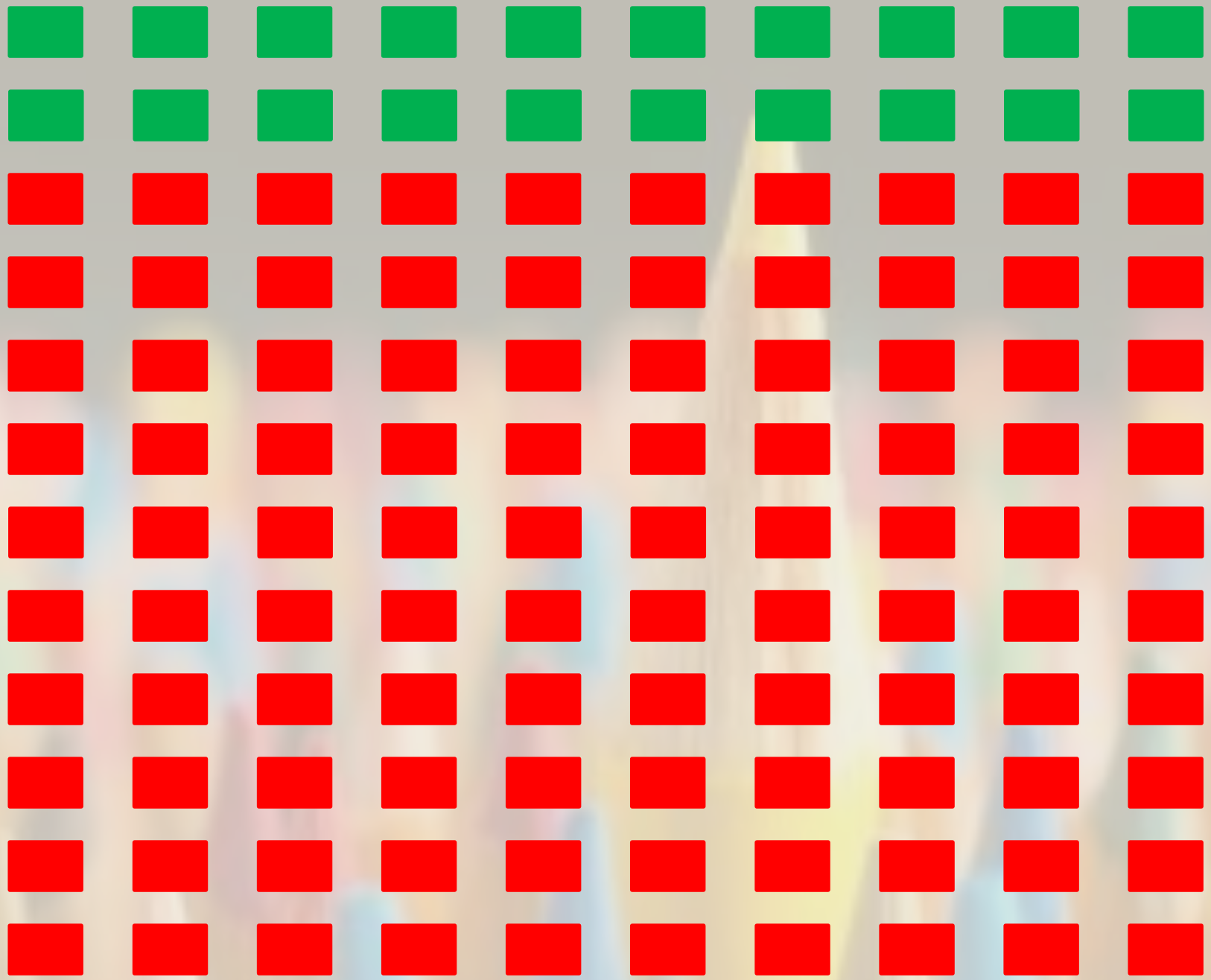
5 Features.
No Dependencies.
120 unique ordering options.

1



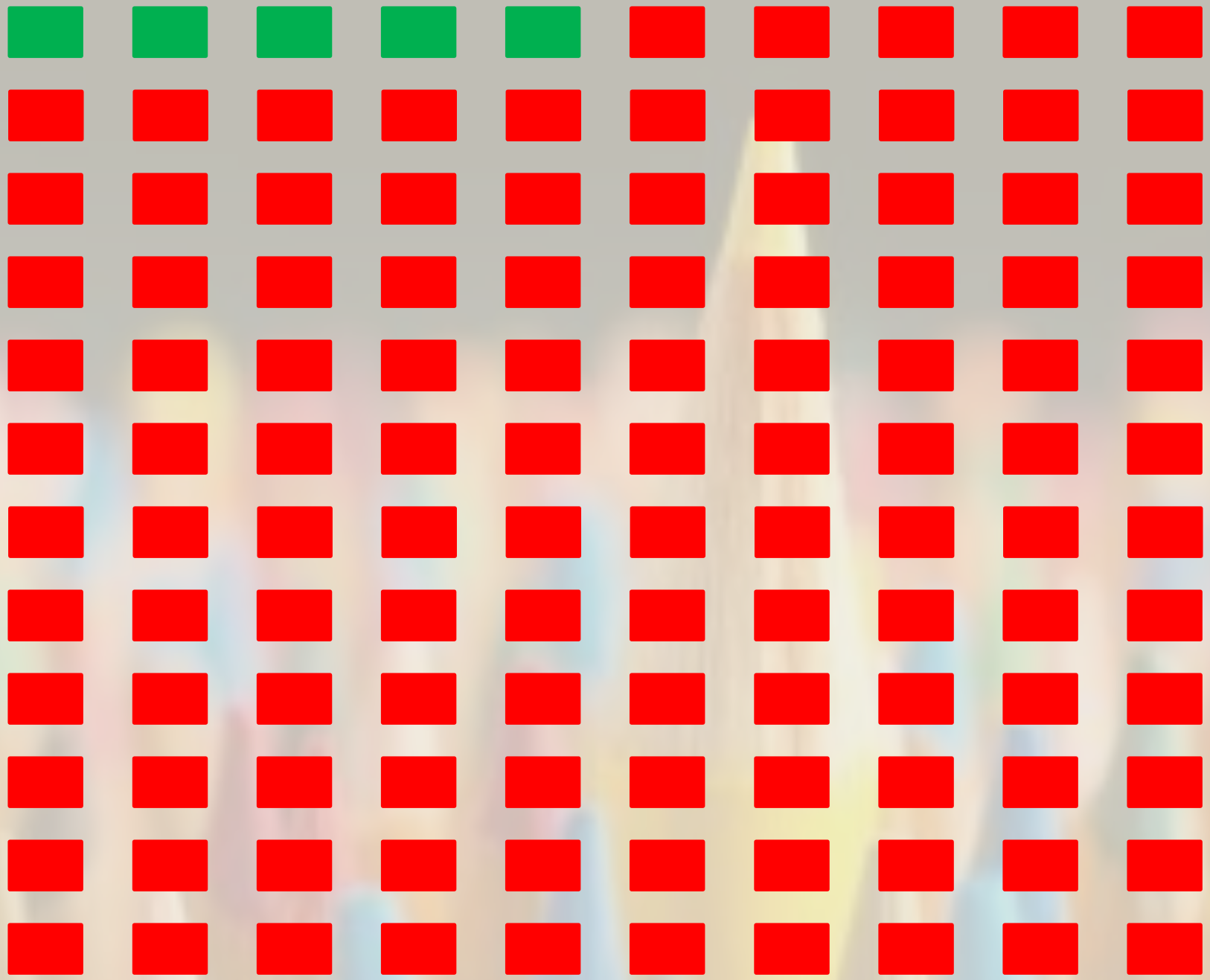
5 Features.
1 Dependency.
60 unique
ordering options.

2



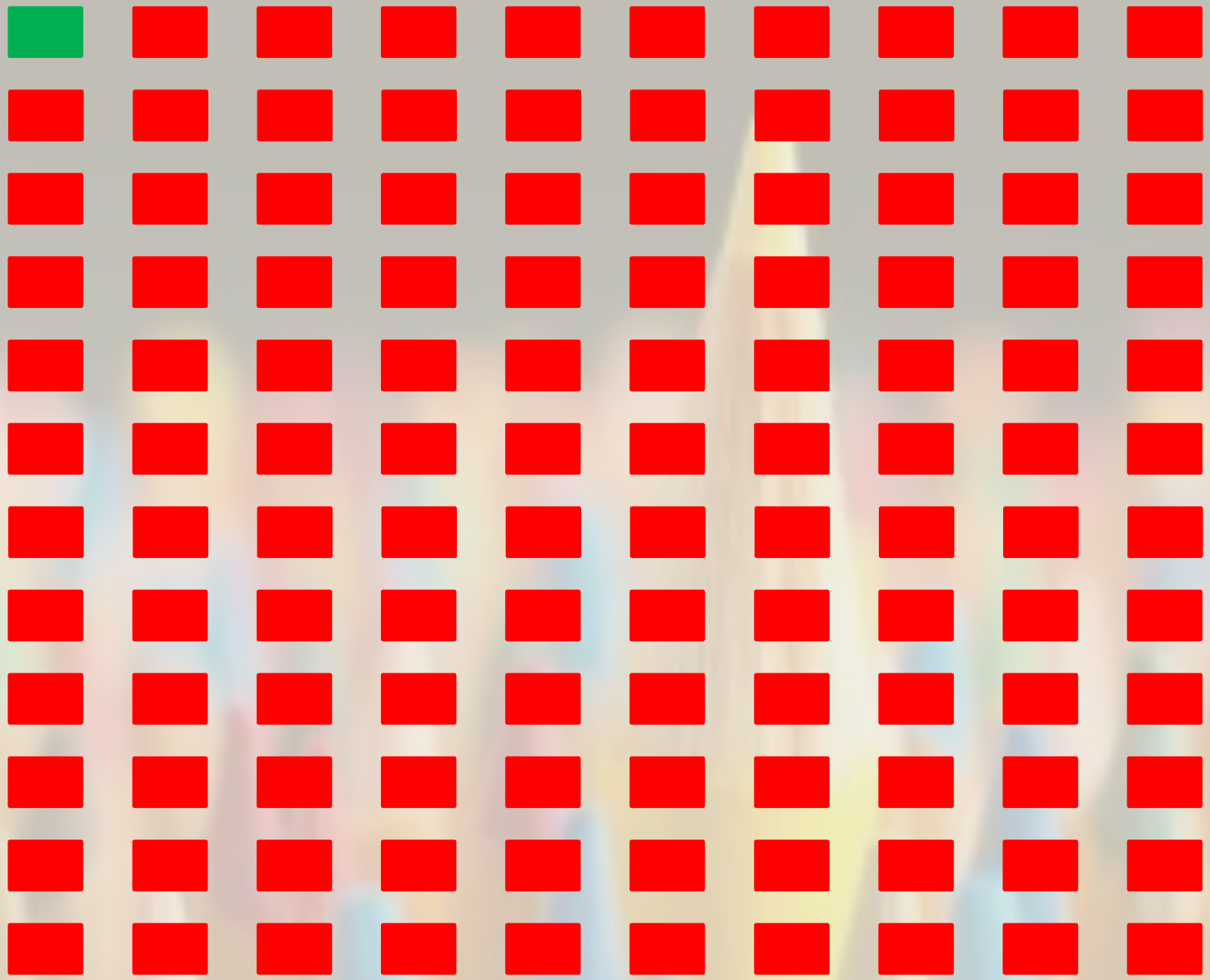
5 Features.
2 Dependencies.
20 unique
ordering options.

3



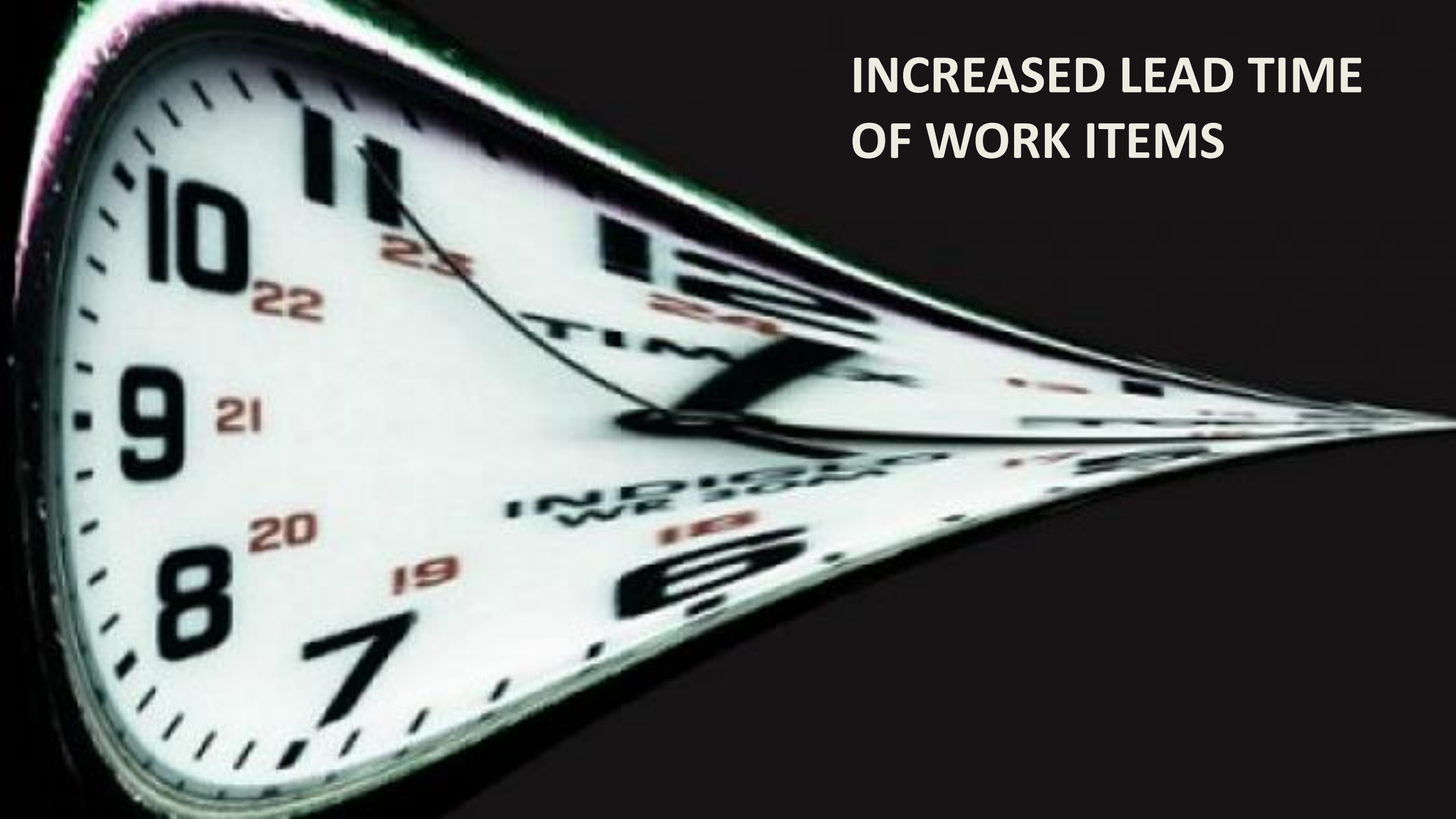
5 Features.
3 Dependencies.
5 unique ordering options.

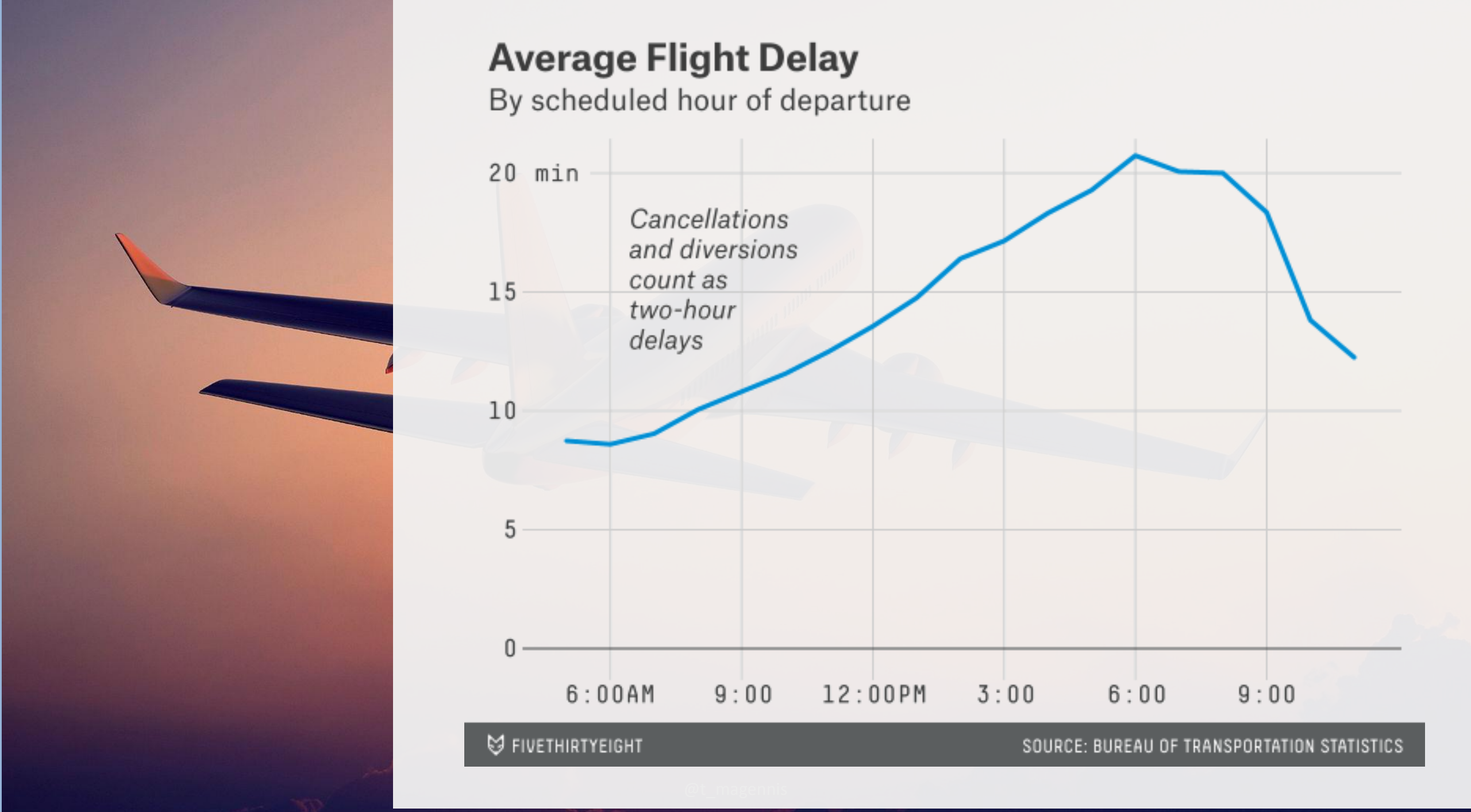
4



5 Features.
4 Dependencies.
1 unique ordering options.

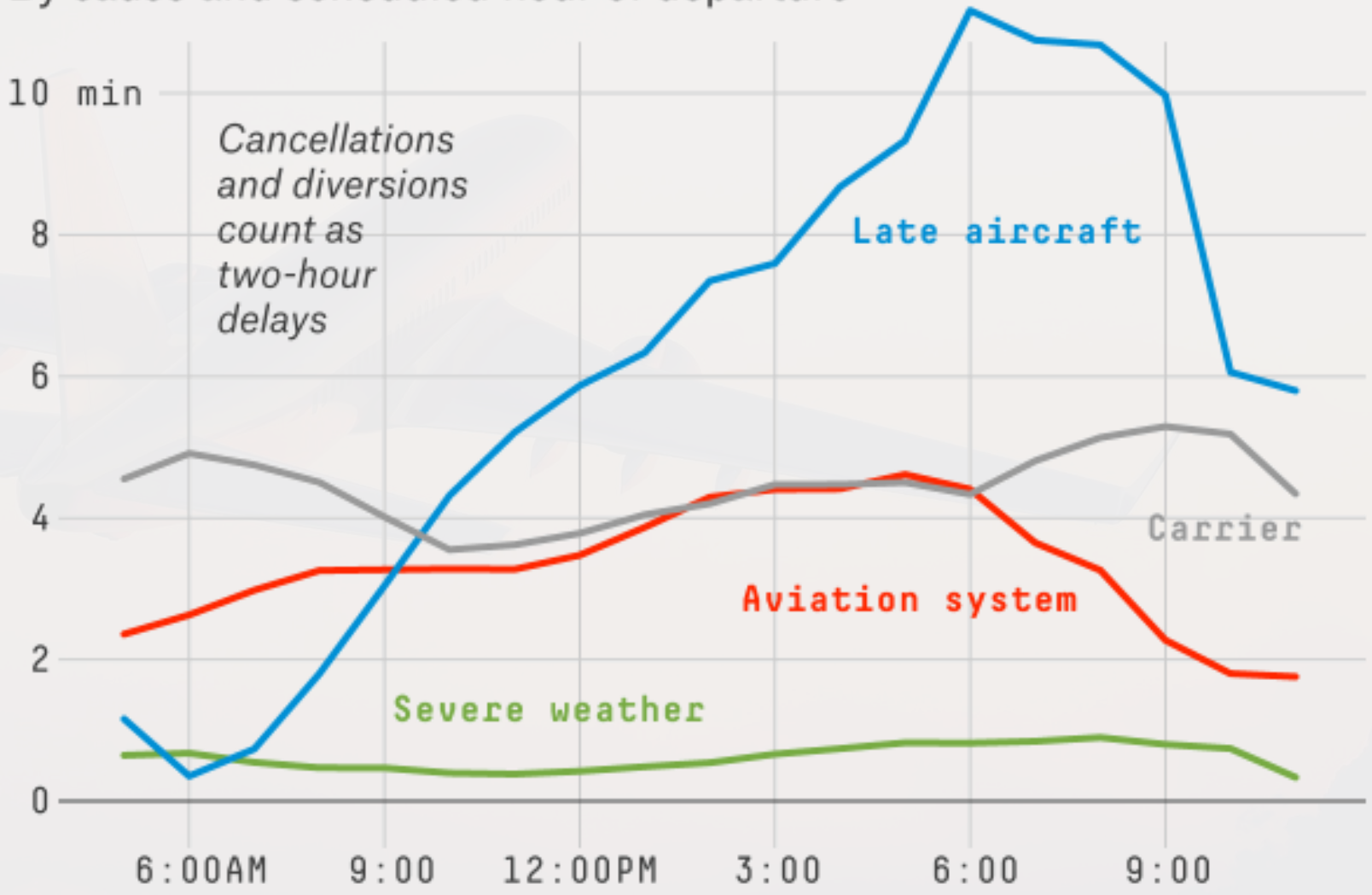
INCREASED LEAD TIME OF WORK ITEMS




Average Flight Delay
By scheduled hour of departure
20 min
15
10
5
0
Cancellations and diversions count as two-hour delays
6:00AM
9:00
12:00PM
3:00
6:00
9:00
FIVETHIRTYEIGHT
SOURCE: BUREAU OF TRANSPORTATION STATISTICS
@t_magennis

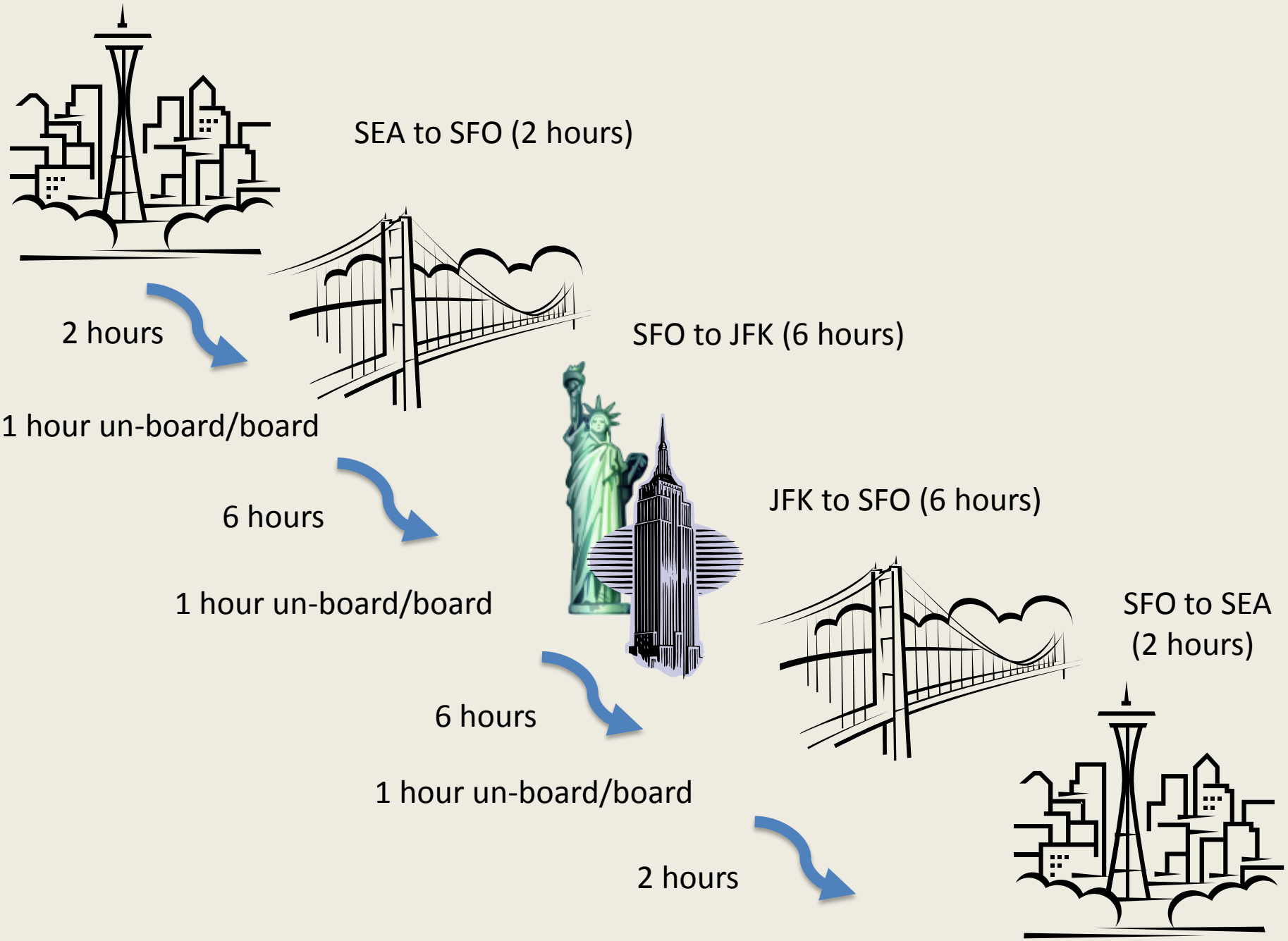
Average Flight Delay

By cause and scheduled hour of departure



FIVETHIRTYEIGHT

SOURCE: BUREAU OF TRANSPORTATION STATISTICS


SEA to SFO (2 hours)
2 hours
SFO to JFK (6 hours)
1 hour un-board/board
6 hours
JFK to SFO (6 hours)
1 hour un-board/board
SFO to SEA
(2 hours)
6 hours
1 hour un-board/board
2 hours

Four people arrange a restaurant booking after work

Q. What is the chance they arrive on-time to be seated?

1 in 16 EVERYONE is ON-TIME

15 TIMES more likely at least on person is late

Person 1	Person 2	Person 3	Person 4

@t_magennis

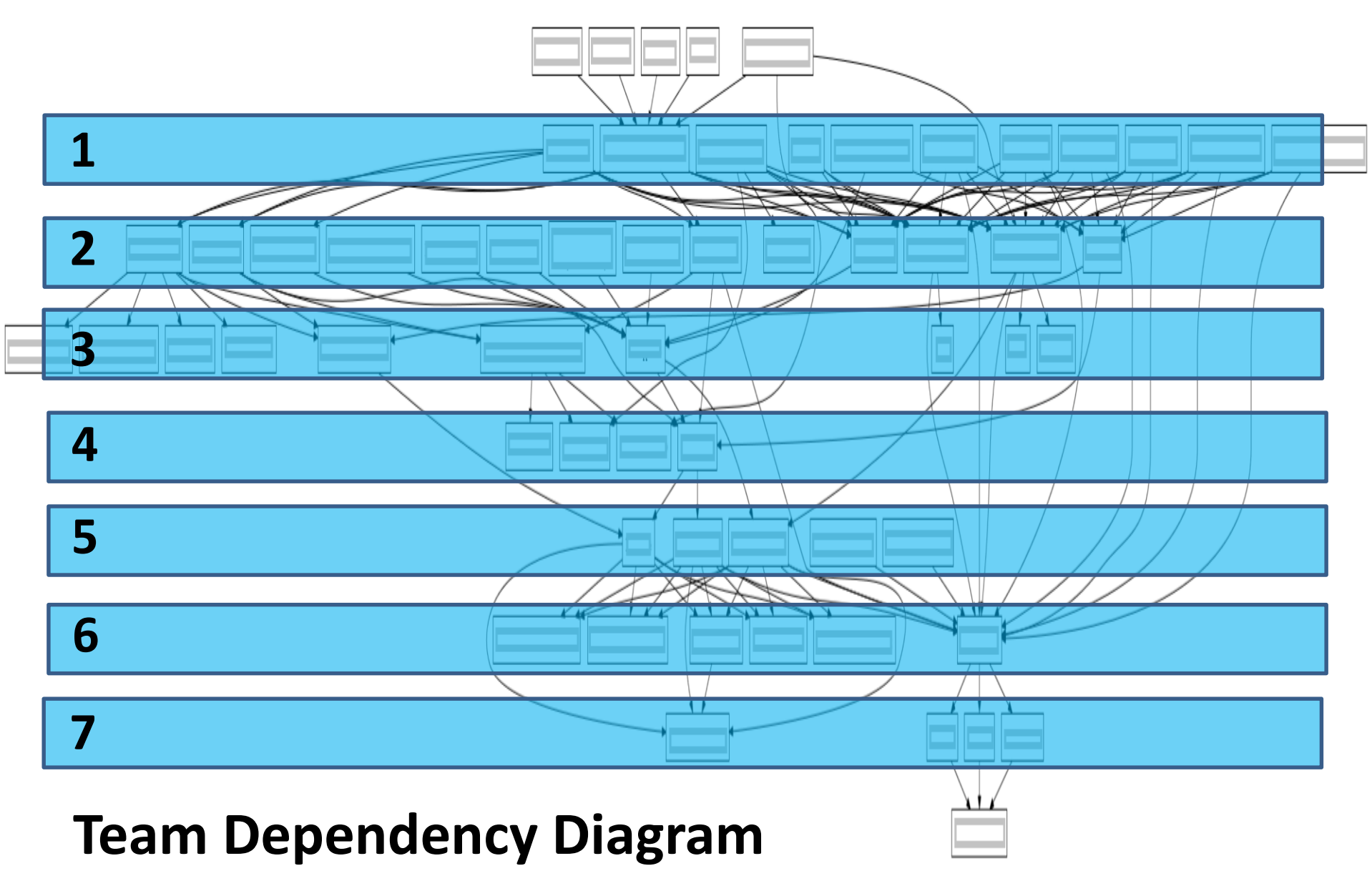


Diagram thanks to Dan Greening (@greening) of SenexRex.

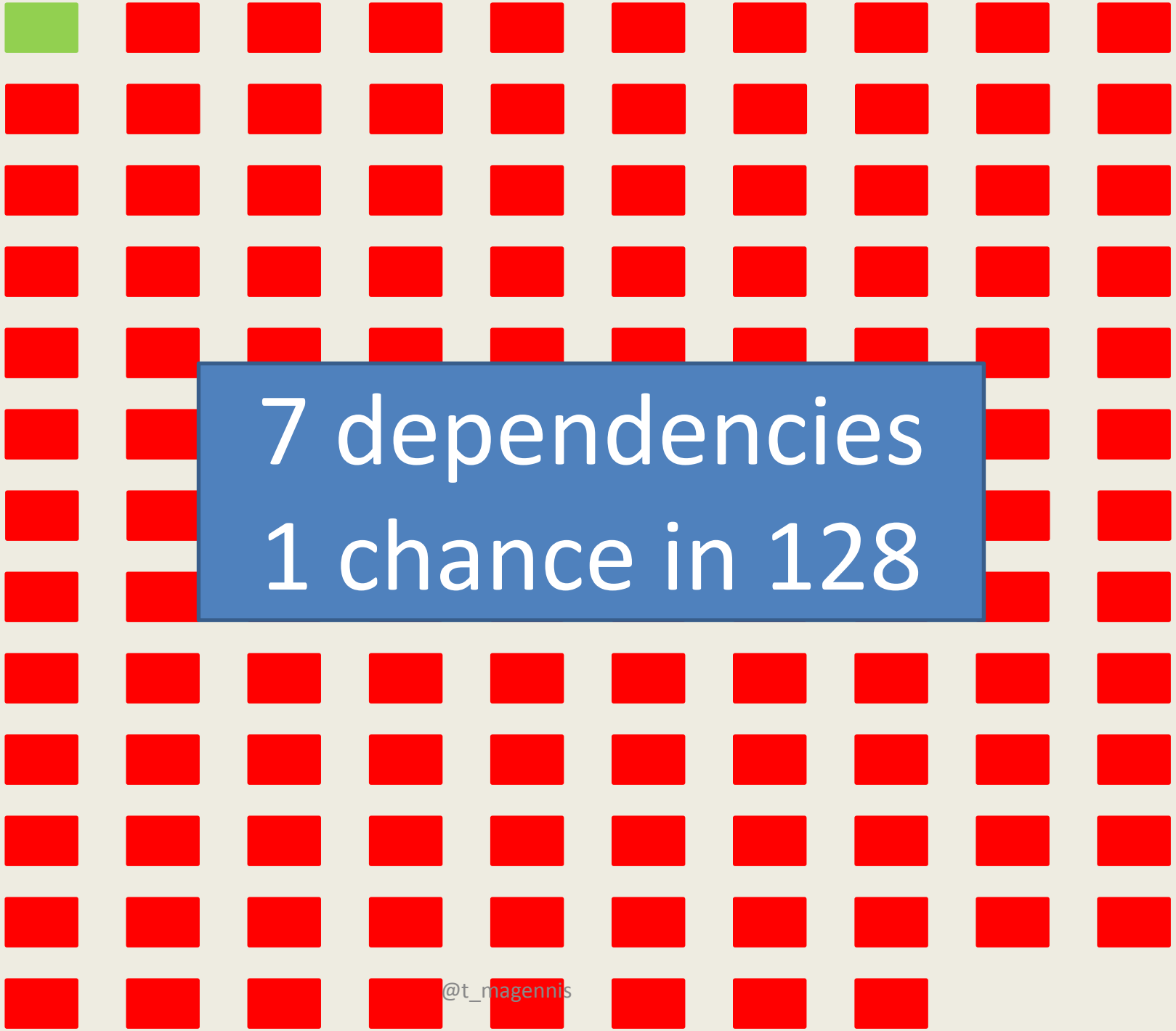
1 in 2^n

or

1 in 2^7

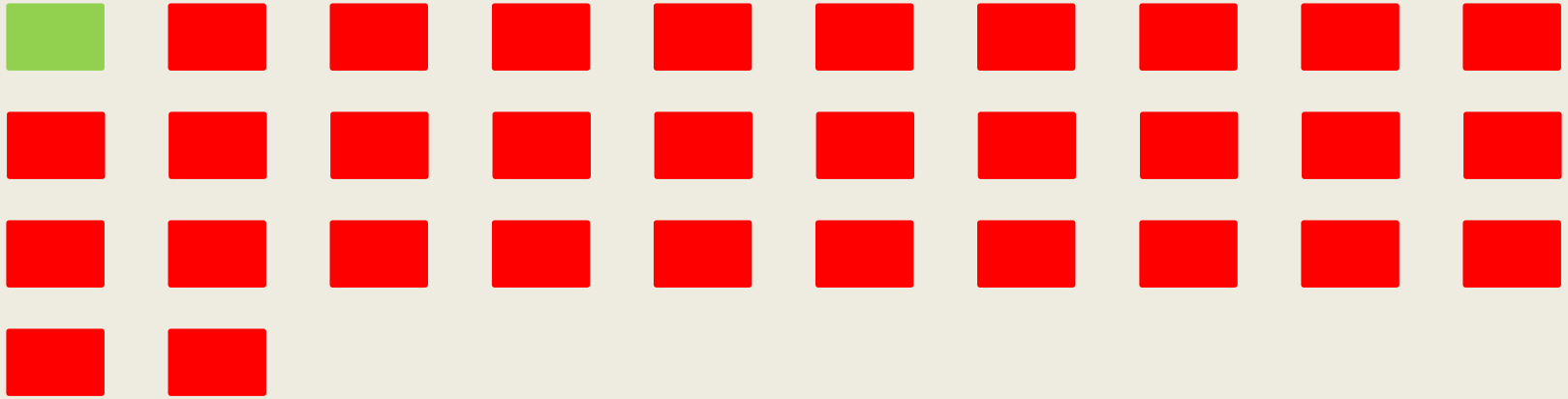
or

1 in 128



7 dependencies
1 chance in 128

6 dependencies
1 chance in 64



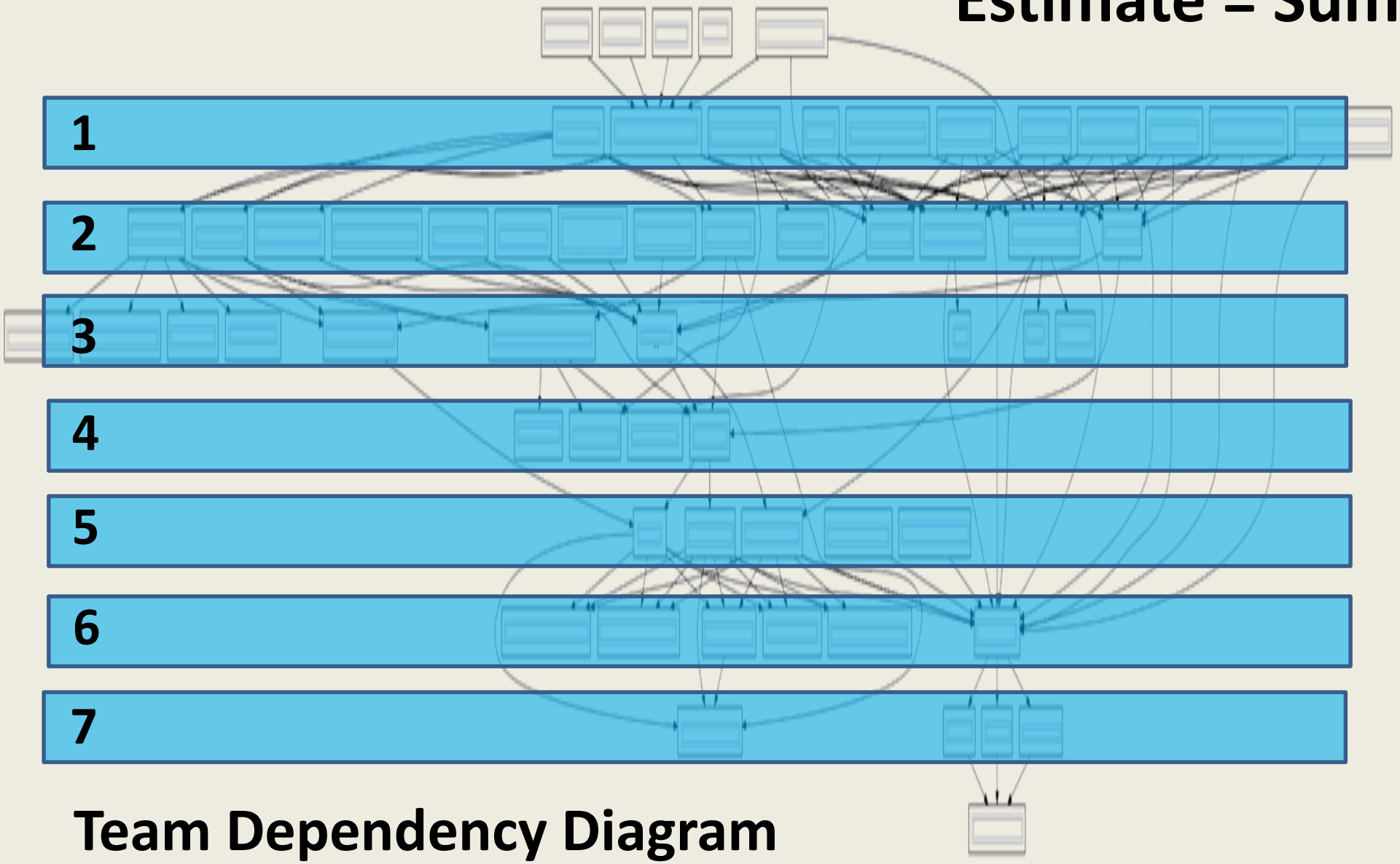
5 dependencies
1 chance in 32

Time = (Optimistic + 4 x Most likely + Pessimistic) / 6

Delays are <u>NOT CAUSED</u> by the item being delayed, its caused by other factors that are unknowable (and unquantifiable) in advance.

It's the start time we struggle to compute, PERT worries about the completion time.

Estimate = Sum Sprints x 1.5

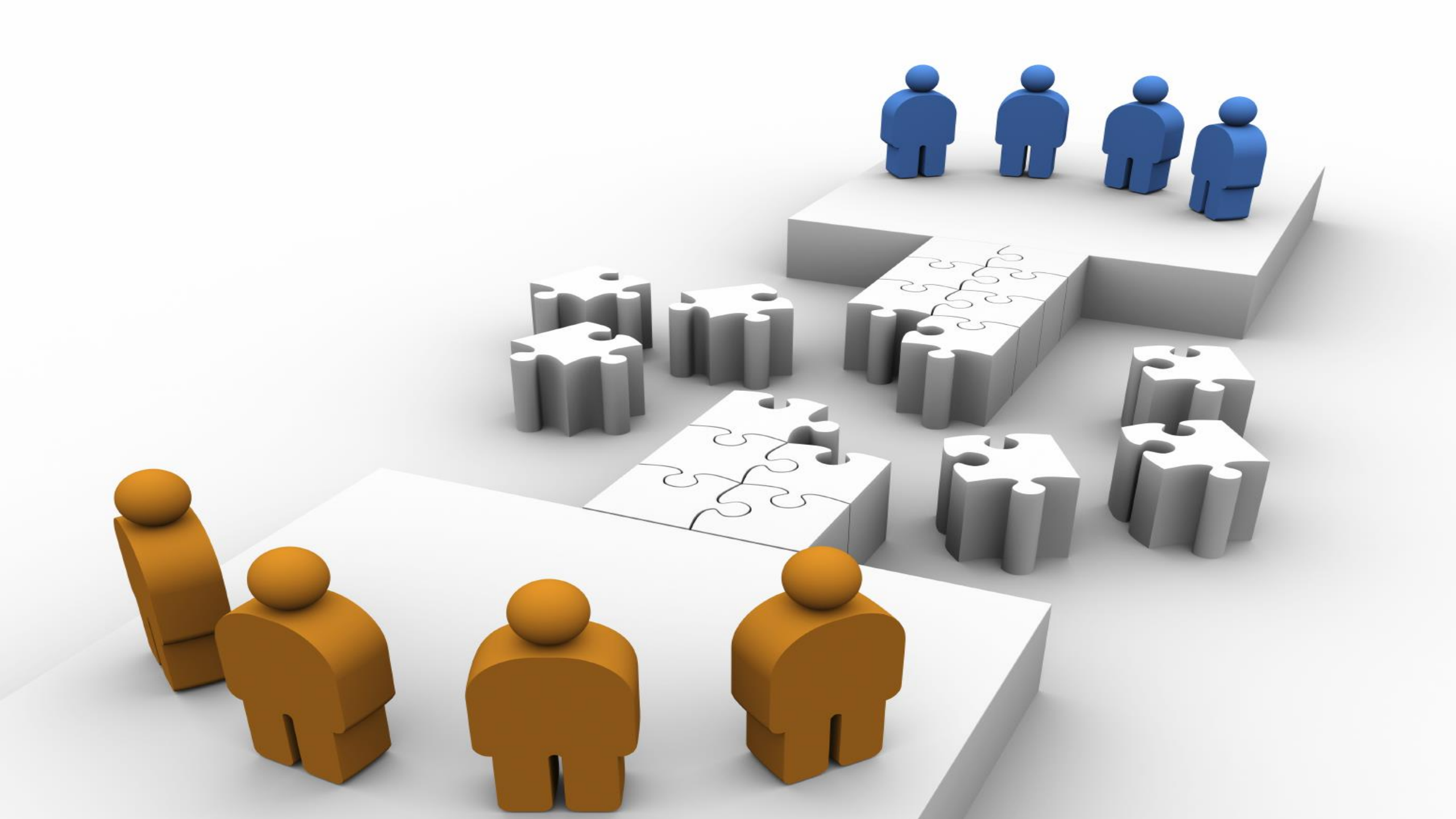


If all work 1 sprint:
7 x 2 x 1.5 = 21 weeks

50% teams will miss the expected time.

Team Dependency Diagram

Diagram thanks to Dan Greening @greening) of SenexRex.

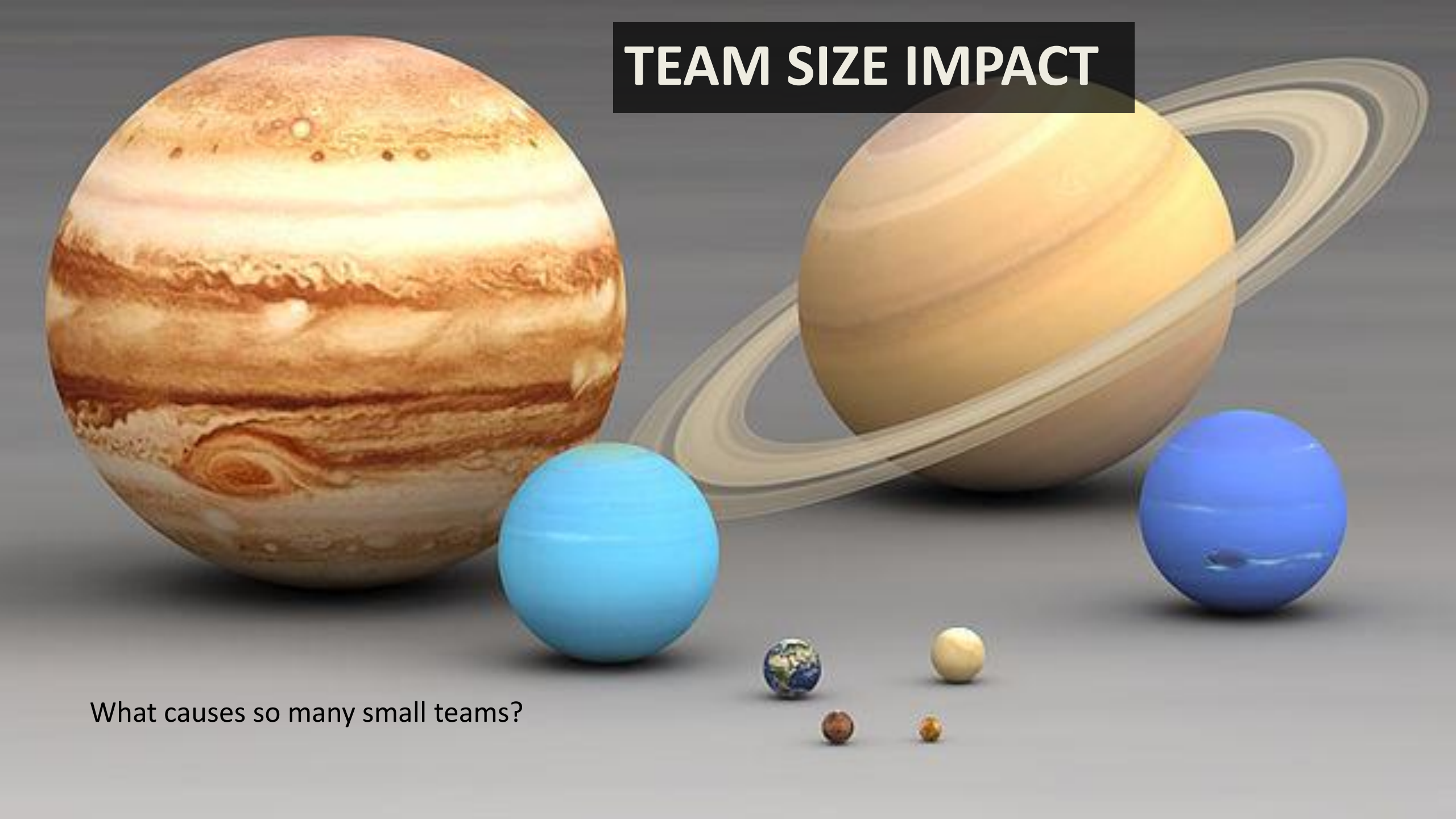


KEEP CALM

IT's only

SOFTWARE TESTING

TEAM SIZE IMPACT

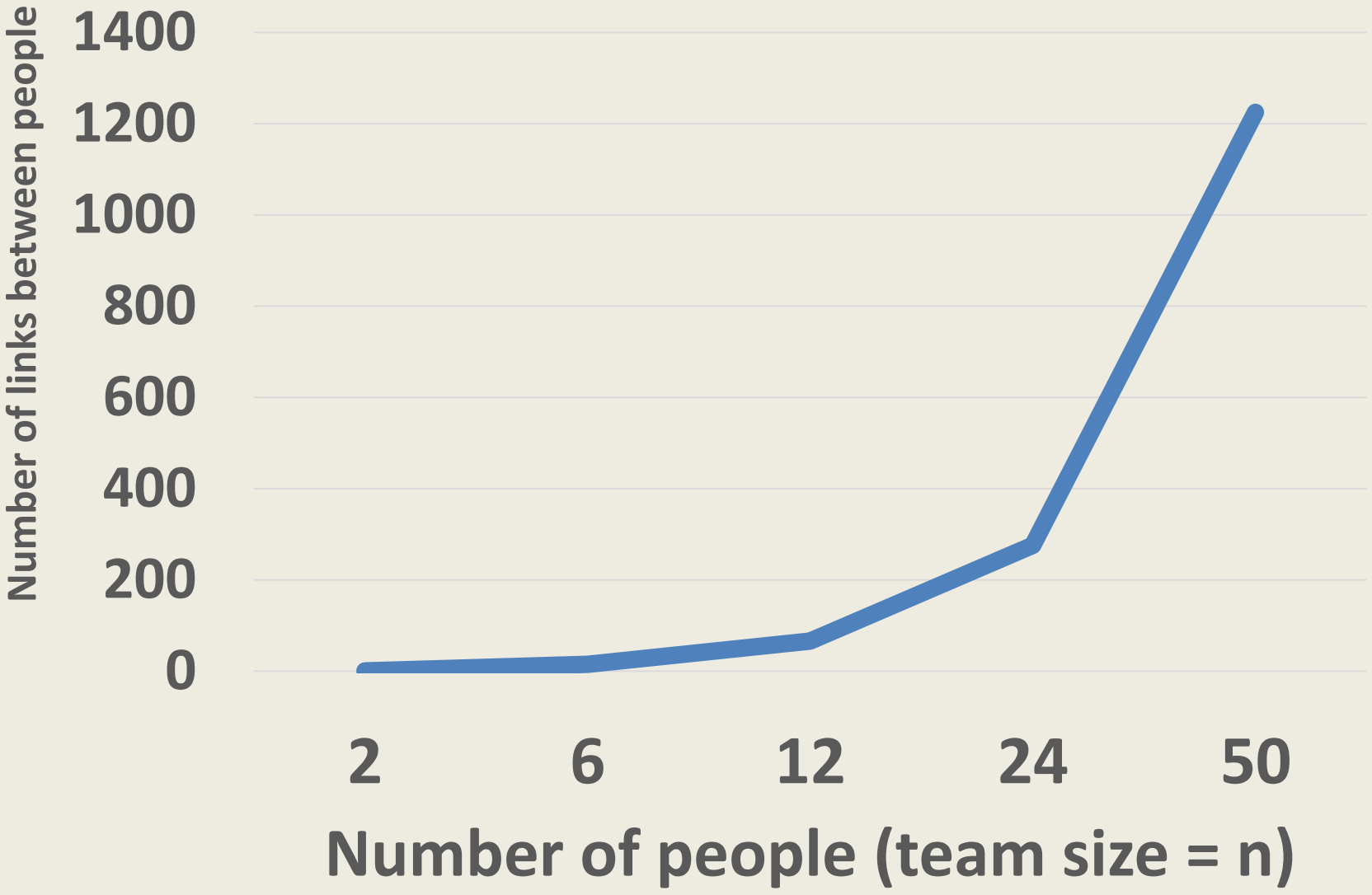


What causes so many small teams?

"The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information"

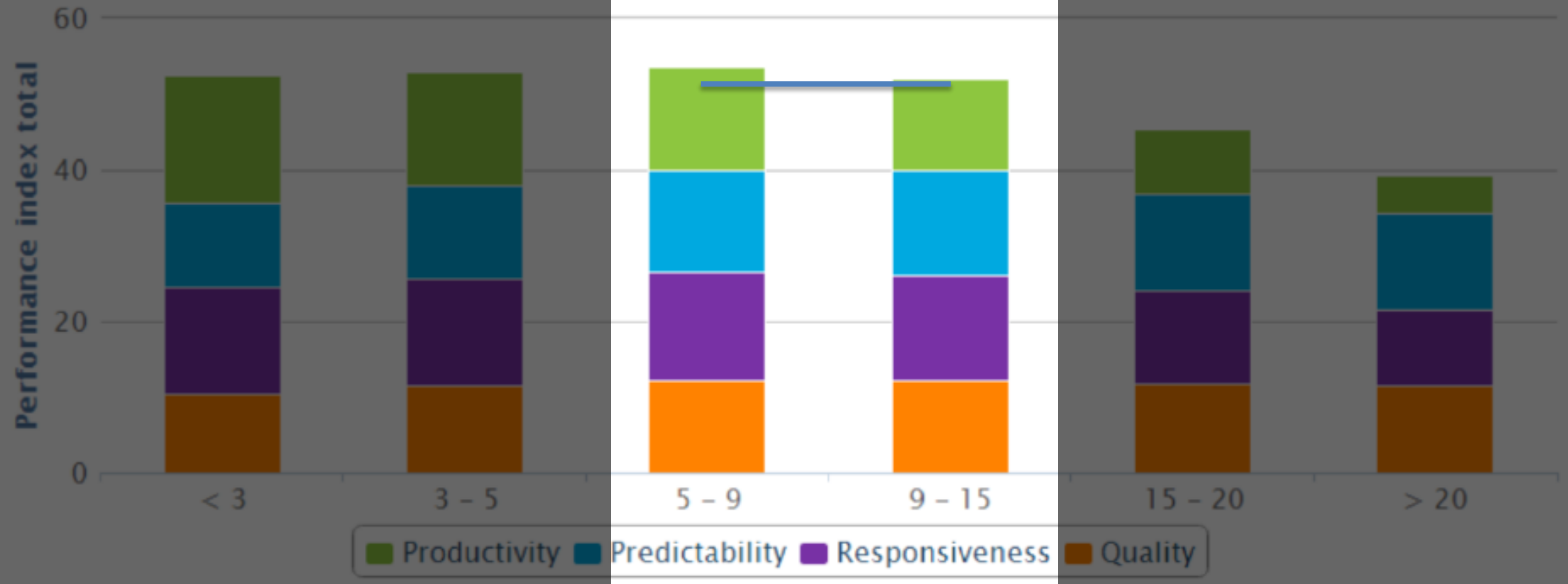
George A Miller – Miller’s Law

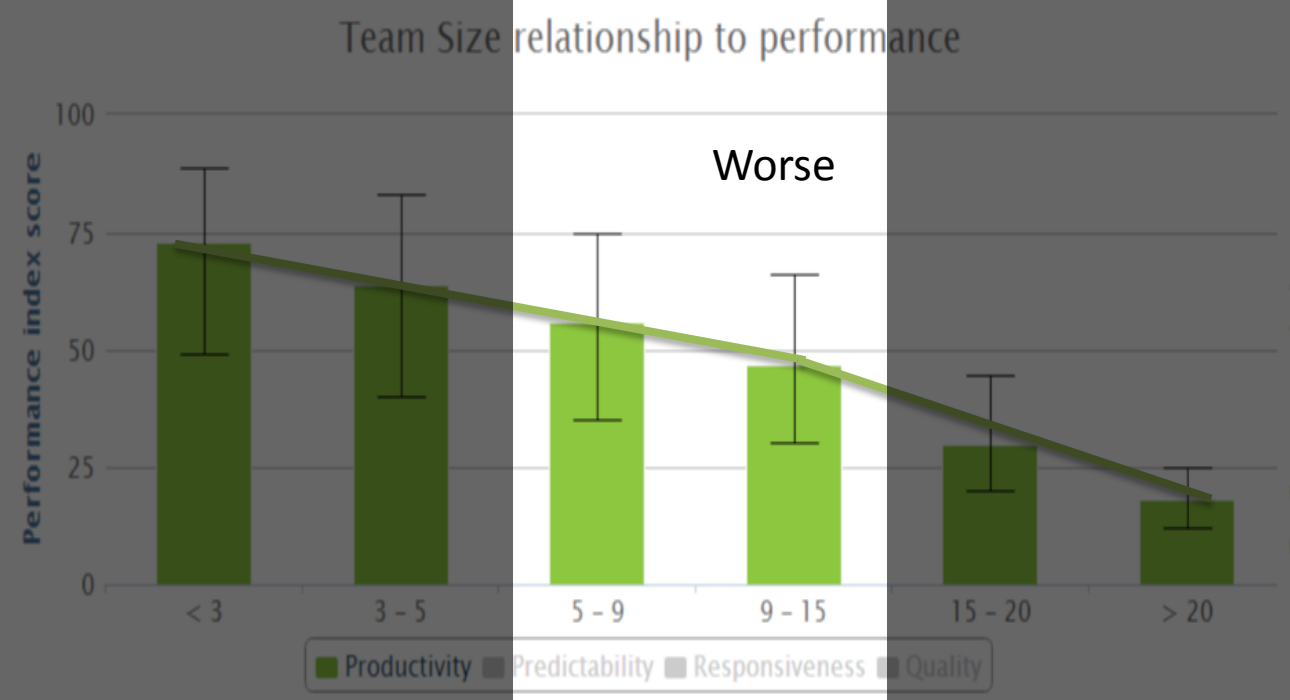
Number of links between people = n x (n-1) / 2

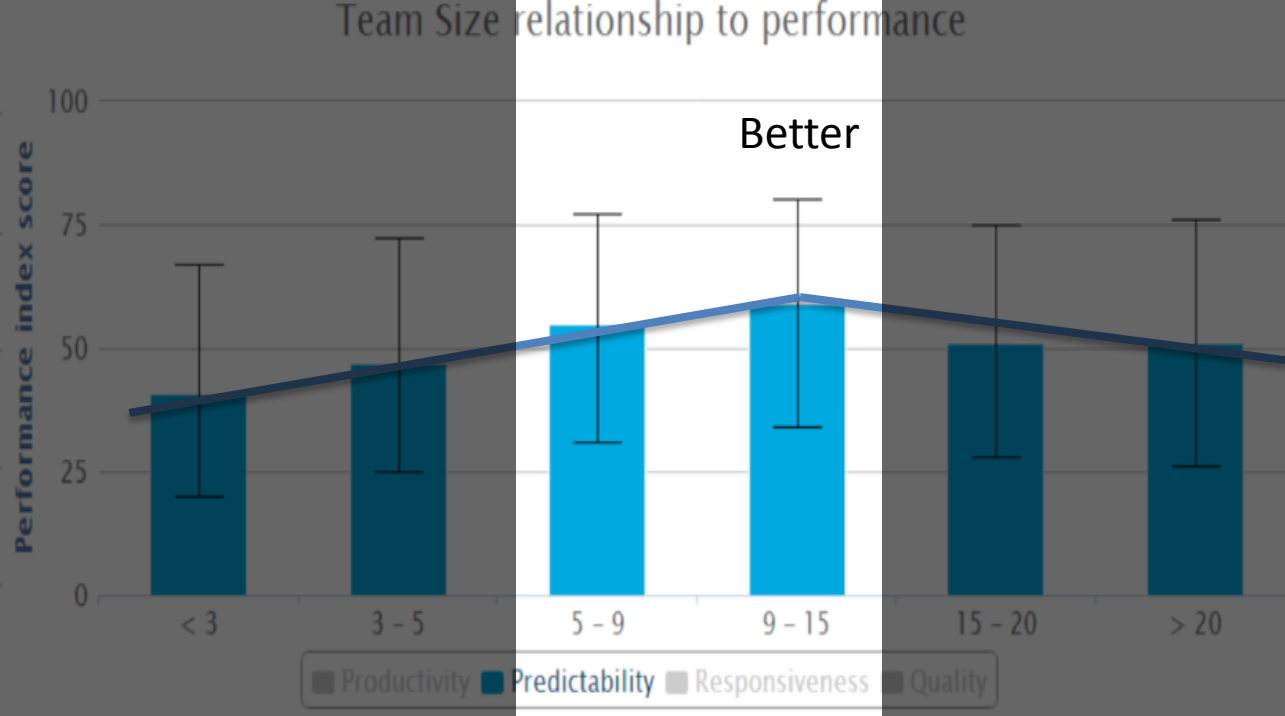


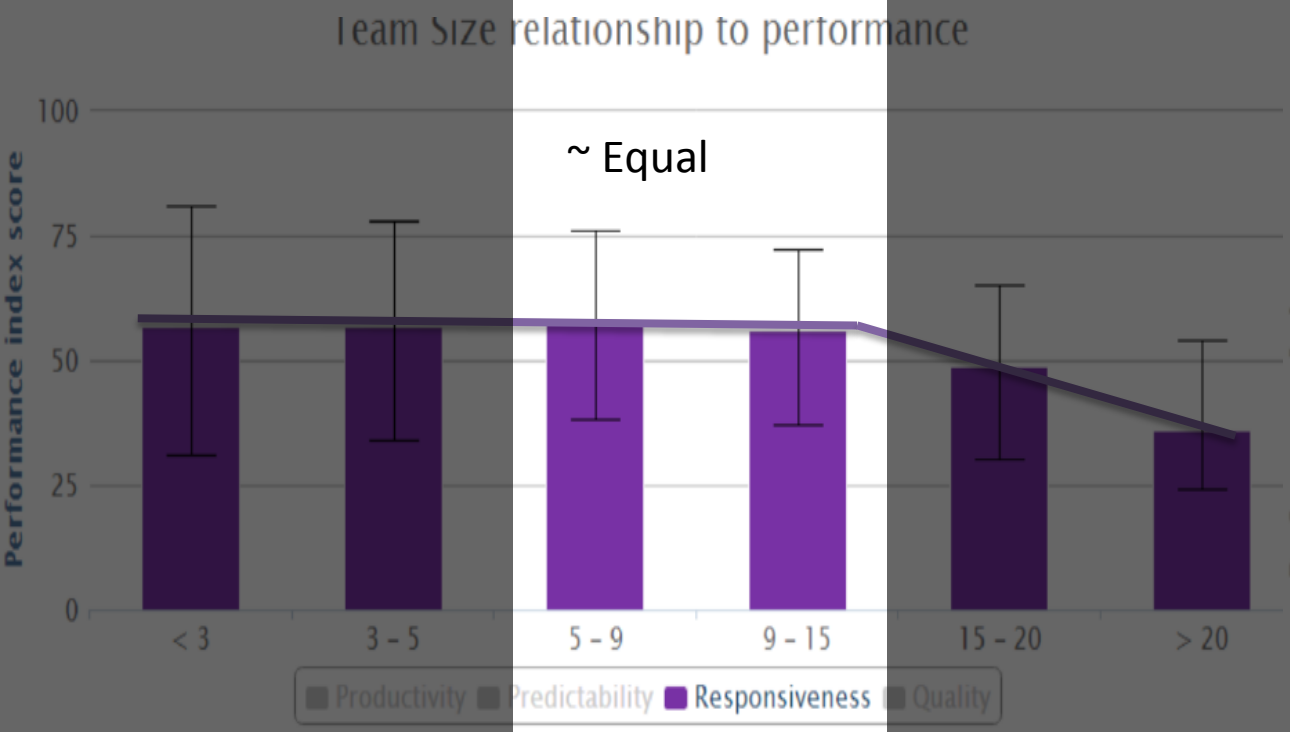
PERFORMANCE

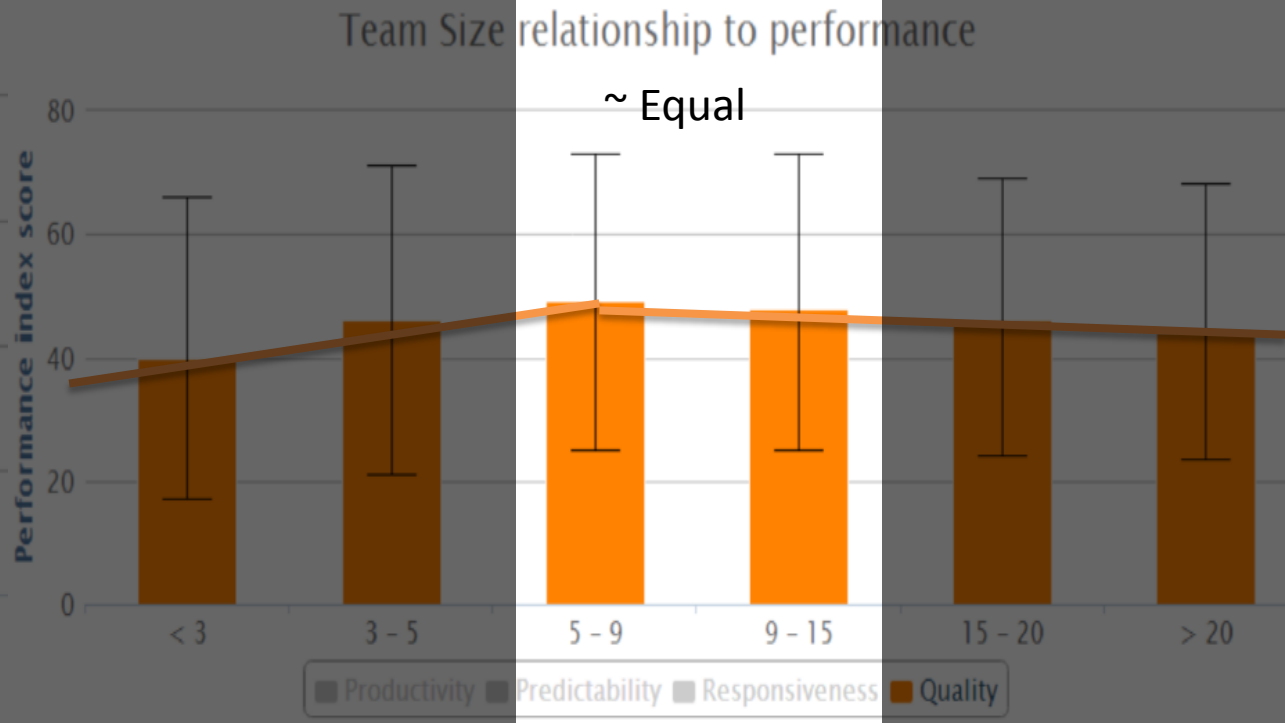
Team Size relationship to performance

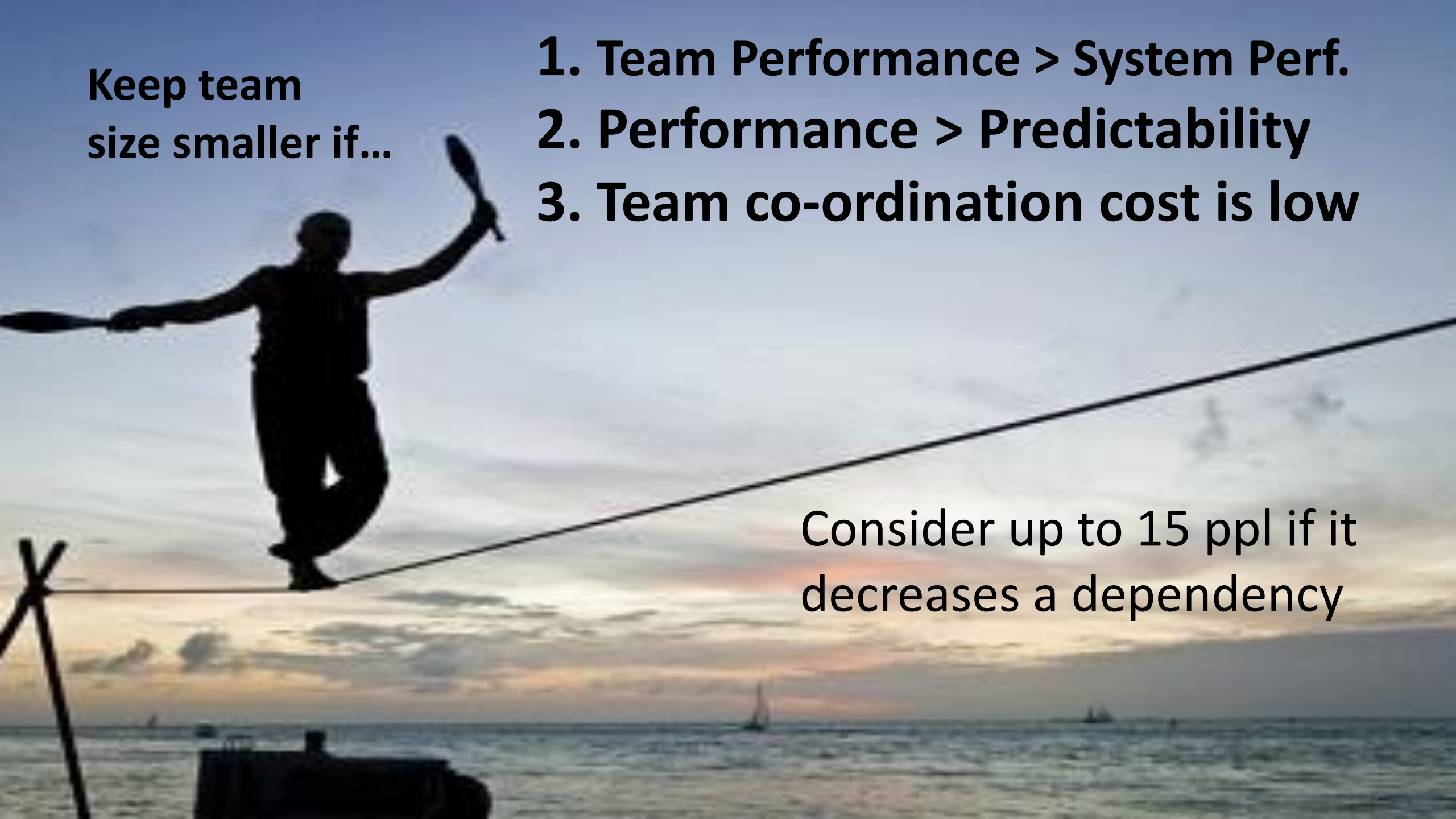



Team Size relationship to performance
Worse
Performance index score
100
75
50
25
0
< 3
3 - 5
5 - 9
9 - 15
15 - 20
> 20
Productivity
Predictability
Responsiveness
Quality


Team Size relationship to performance
Better
Performance index score
100
75
50
25
0
< 3
3 - 5
5 - 9
9 - 15
15 - 20
> 20
Productivity
Predictability
Responsiveness
Quality


Team Size relationship to performance
~ Equal
Performance index score
100
75
50
25
0
< 3
3 - 5
5 - 9
9 - 15
15 - 20
> 20
Productivity
Predictability
Responsiveness
Quality


Team Size relationship to performance
~ Equal
Performance index score
80
60
40
20
0
< 3
3 - 5
5 - 9
9 - 15
15 - 20
> 20
Productivity
Predictability
Responsiveness
Quality



Keep team size smaller if…

1. **Team Performance > System Perf.**
2. **Performance > Predictability**
3. **Team co-ordination cost is low**

Consider up to 15 ppl if it decreases a dependency

TEAM ORGANIZATION DESIGN OPTIONS

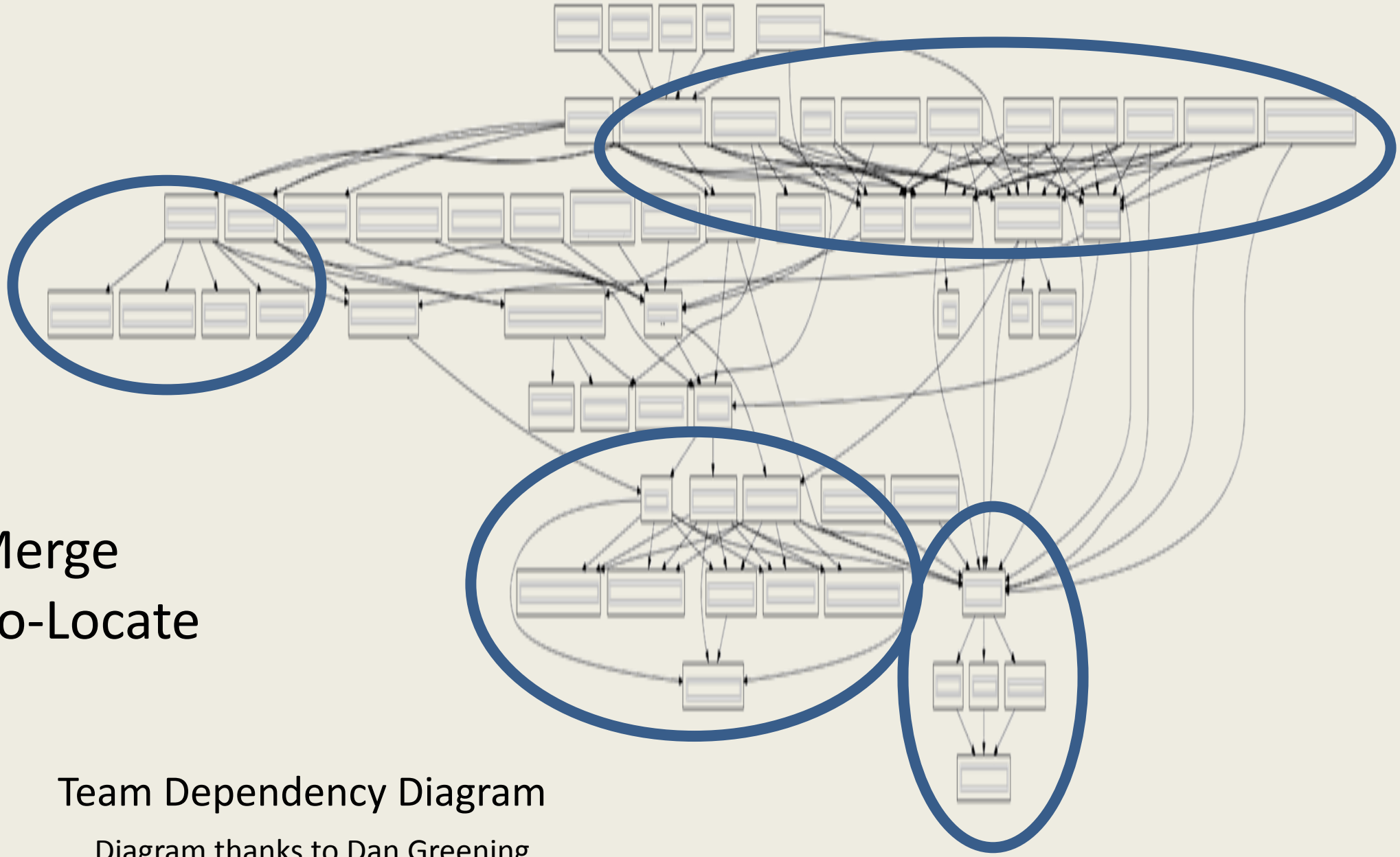
Reducing Dependencies: Team Structure Options

Merge teams

Create multi-disciplined feature teams

versus

Co-locate teams who depend on each other

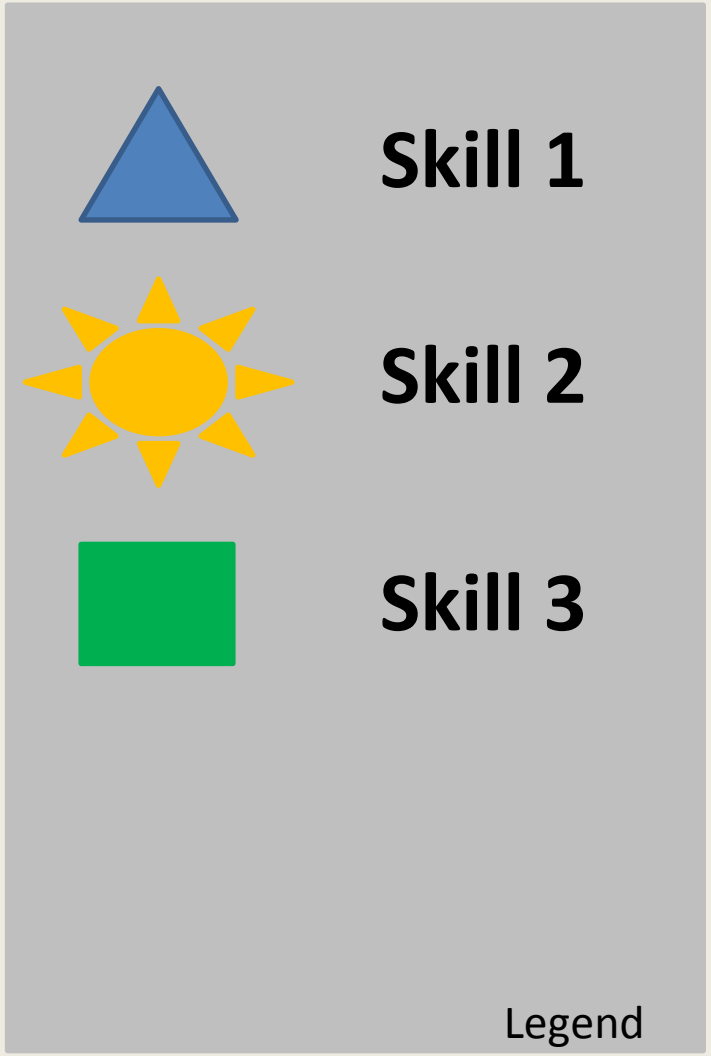


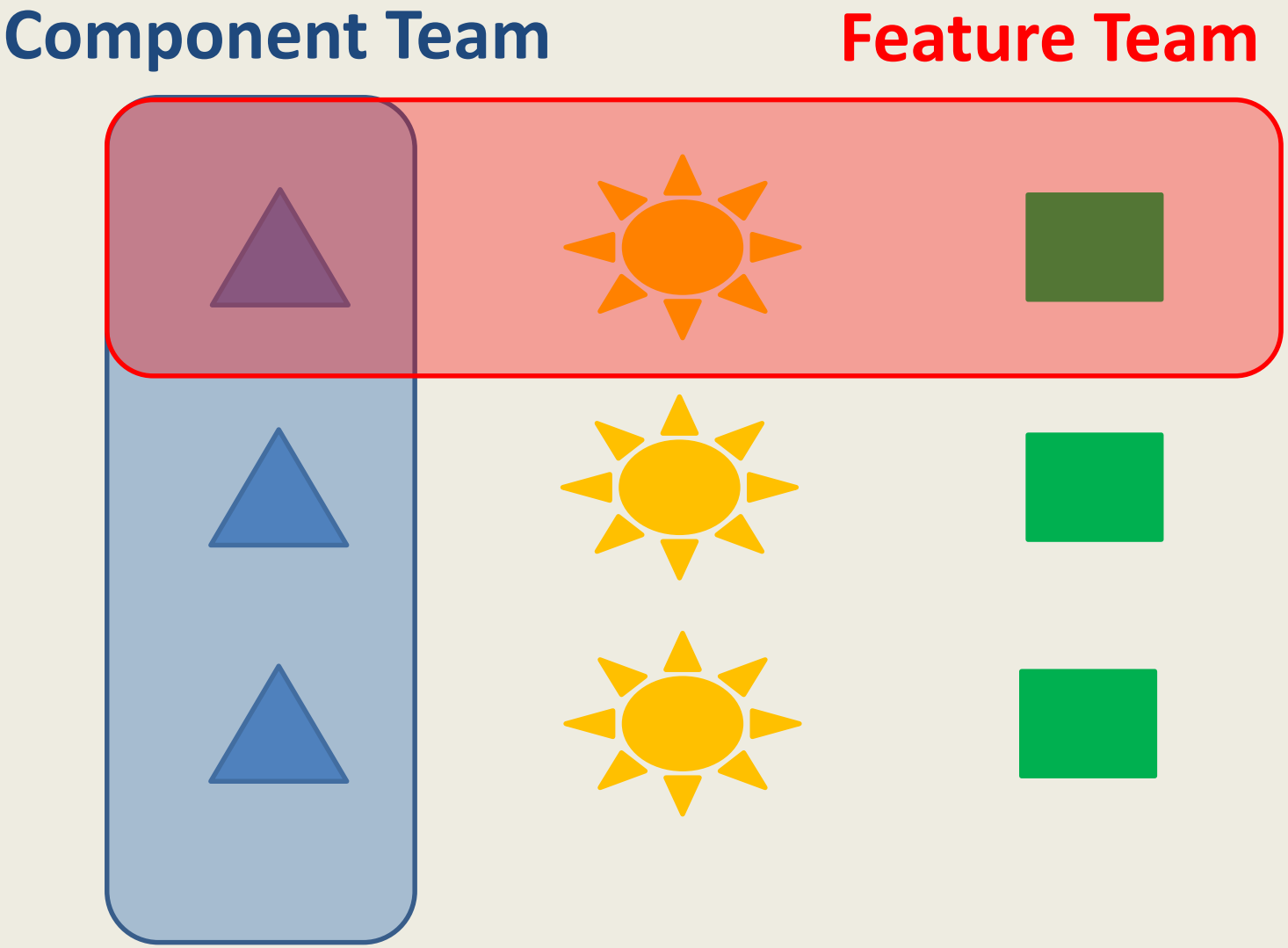
1. Merge
2. Co-Locate

Team Dependency Diagram

Diagram thanks to Dan Greening (@greening) of SenexRex.

Matrix – Component vs Feature





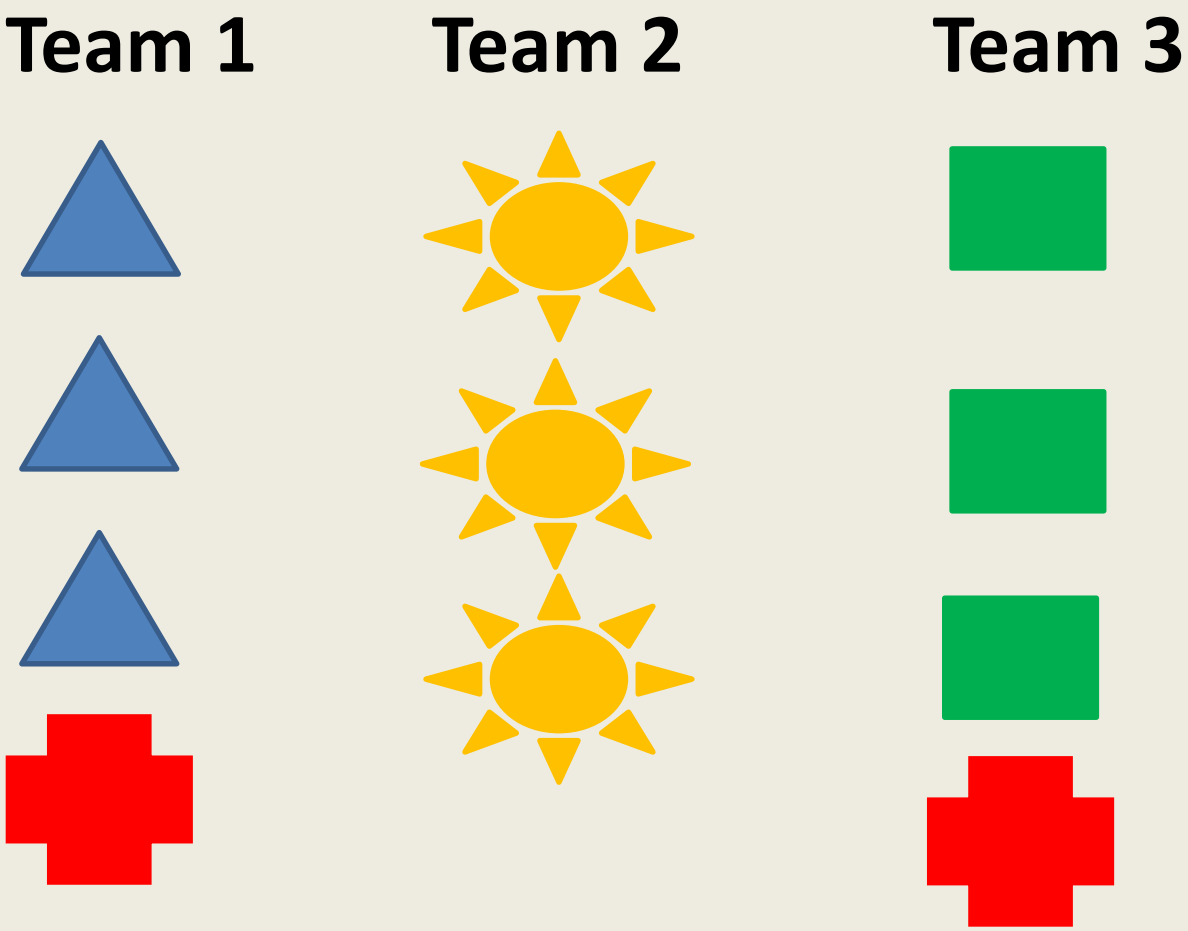
Component or Skillset Area Teams

Pros

- Consistent practices
- Predictable in isolation
- Complete area knowledge
- Fast ramp-up of new members

Cons

- Many dependencies
- Low predictability as a system
- No system understanding



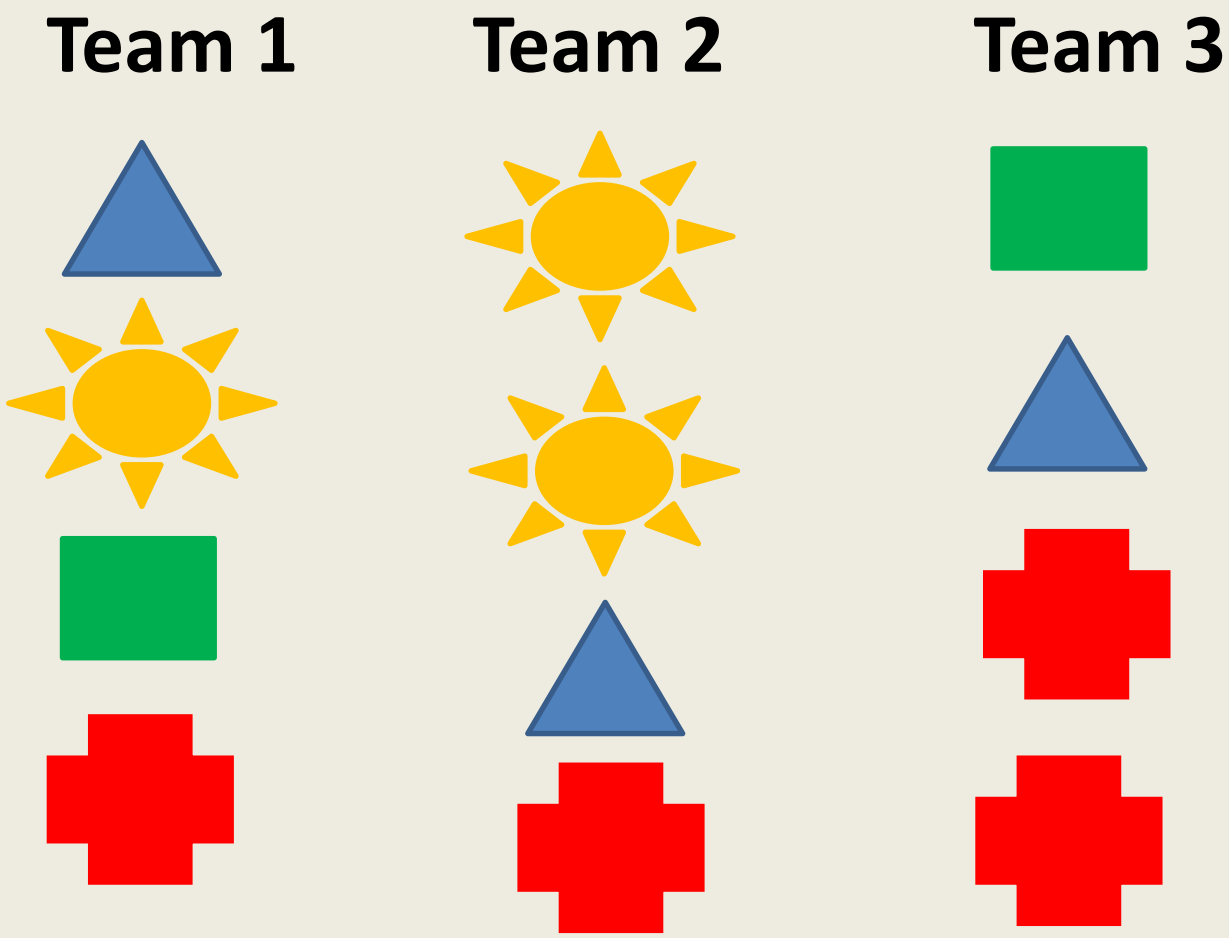
Feature Teams

Pros

- Few/Any Dependencies
- Predictable feature delivery
- Complete feature knowledge

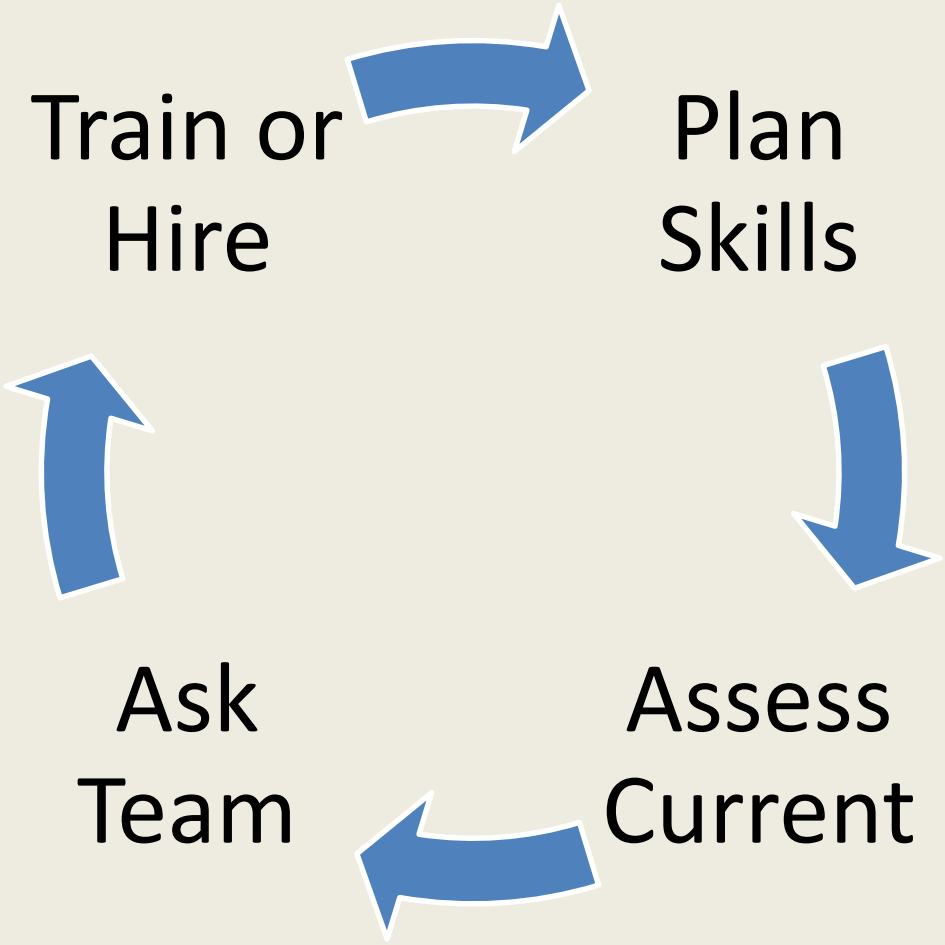
Cons

- Divergent practices
- Code / build Integration harder
- Beware of single “expert”



Planning and Growing Teams

- Skill falls into levels
  1. Can teach others
  2. Can do
  3. Have the desire
  4. No idea, and never will!
- Continuous Cycle
  - Role of managers
    - Capability
    - Risk Exposure

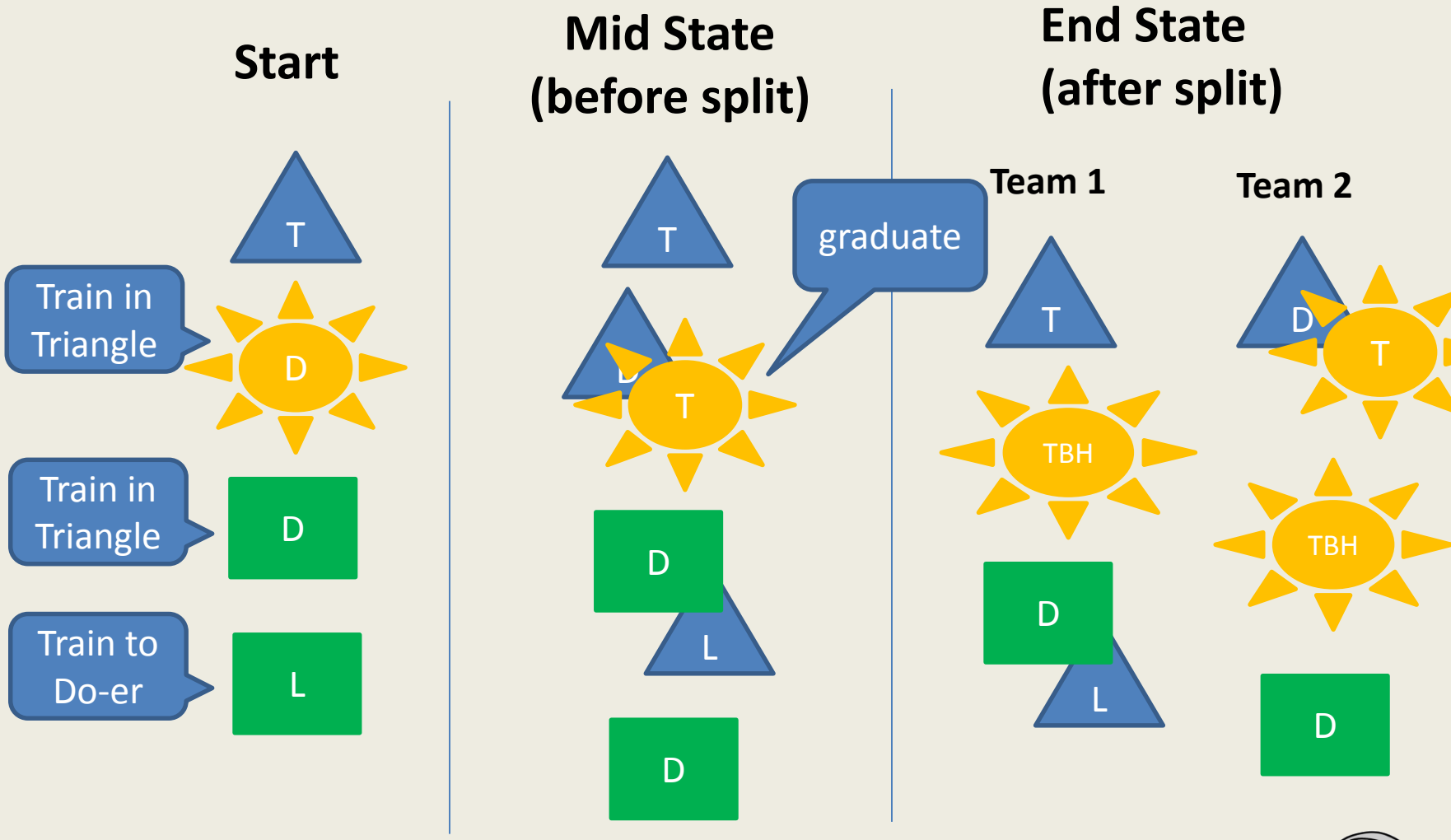


Growing Teams – Skills != People

4 ppl	Skill 1	Skill 2	Skill 3
Teacher	1	0	0
Do-er	0	1	1
Learner	0	0	1

4 ppl	Skill 1	Skill 2	Skill 3
Teacher	1	1	0
Do-er	1	0	2
Learner	1	0	0

6 ppl	Skill 1	Skill 2	Skill 3
Teacher	1	1	0
Do-er	1	1	2
Learner	1	1	0



Skill Assessment – Knowing and Growing

Download from: **Bit.ly/SimResources** (Spreadsheets, Capability Matrix.xlsx)

Capability Survey For Printing (enter skills in the Settings worksheet)

Team Name: _____ Your Name: _____

For each capability choose from the list of CURRENT skill level values. If in doubt, err low (left)!

	Know nothing	Can run and use the tools needed	Can tweak it or do easy bug fixes	Can start from nothing and create	Expert level
CSS	☐	☐	☐	☐	☐
Javascript	☐	☐	☐	☐	☐
DB Backup/Restore	☐	☐	☐	☐	☐
	☐	☐	☐	☐	☐

For each skill, choose from the list of DESIREABLE values. If in doubt, err high (right)!

	I'd quit rather than do this...	Actively Avoid, unless coerced...	Willing to learn	Strongly Interested	Please, Please, Please...
CSS	☐	☐	☐	☐	☐
Javascript	☐	☐	☐	☐	☐
DB Backup/Restore	☐	☐	☐	☐	☐
	☐	☐	☐	☐	☐

	A	B	C	D
1		CSS	Javascript	DB Backup/Restore
2	Person 1	Can run and use the tools needed	Know nothing	Can run and use the tools needed
3	Person 2	Know nothing	n start from nothing and create	Can tweak it or do easy bug fixes
4	Team 1		ow nothing	Can start from nothing and create
5				
14				
15	**Analysis:**			
16		CSS	Javascript	DB Backup/Restore
17	Teach & Create	1	1	1
18	Do & Maintain	1	1	2
19	Novice & Learner	1	0	1
20				

Know nothing
Can run and use the tools needed
Can tweak it or do easy bug fixes
Can start from nothing and create
Expert level

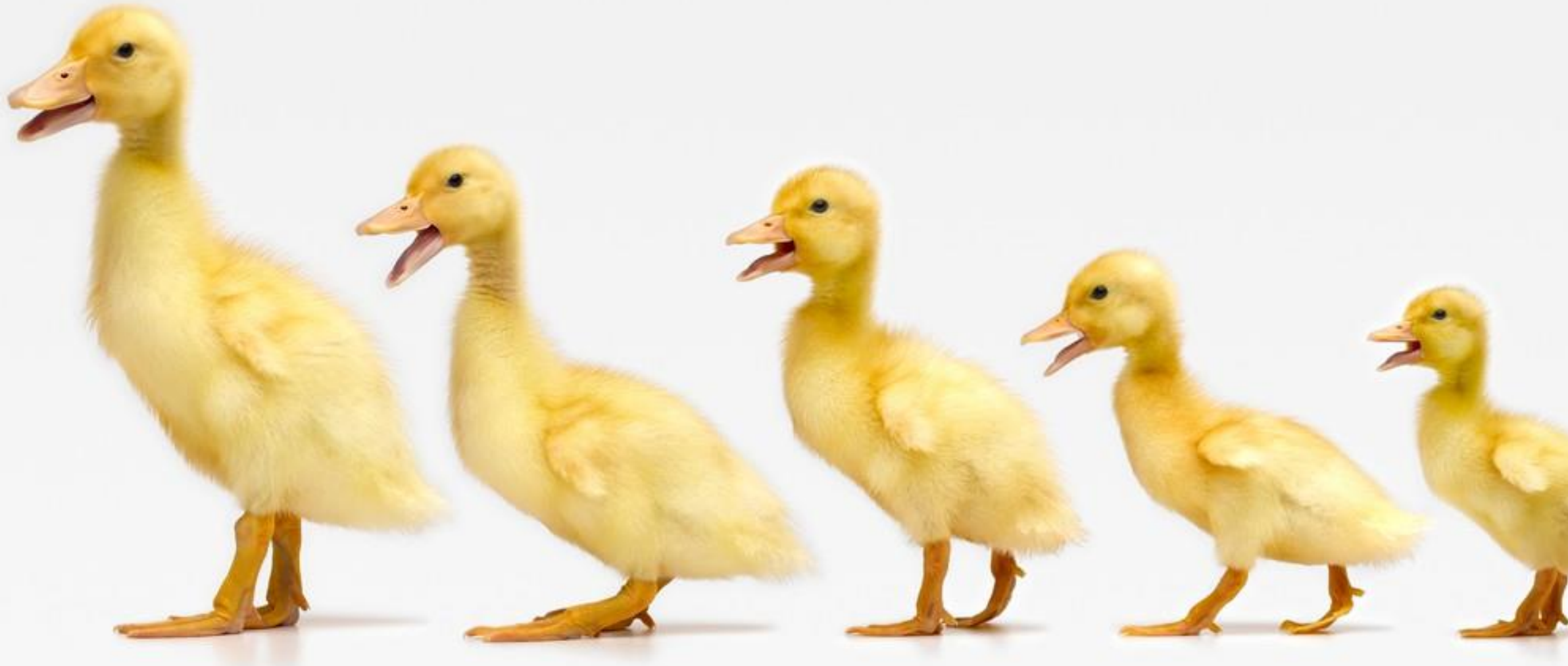
21 **General guidelines: 0 = bad, 1 = single point of failure, >2 cool!**

22

23 **Teach & Create:** These are the people/teams who can create new work and teach others. You need at least one (right?). Are you able to cope if that person i
24 off sick or vacation? If not, then train up a maintainer or bench employee?
25 **Do & Maintain:** These are the people/teams who can maintain current work, but struggle to create new work. If new work isn't expected, it may be ok to
26 have no captains but a crew of maintainers. Still, one seems too risky? Grow from the bench.
27 **Novice & LEarner (bench):** These are the people/teams who although haven't got this skill yet, have the tools required to perform this task if mentored or
28 paired with a Player. If you have too few Captains and Players, you need to develop these urgently.
29

Download from: Bit.ly/SimResources (Spreadsheets, Capability Matrix.xlsx)

CLEAR AND ALIGNED PRIORITIES



Incentives

Thanks to Lisa Long and Chris Matts for the cookie analogy.

Different players have different rewards.

“Do a feature, get a cookie”

~ Lisa Long

If rewards aren’t aligned, different ordering decisions are made

Automatic Cookies Aligning Machine Wit

FOB Price:	Get Latest Price
Min.Order Quantity:	1 Set/Sets
Supply Ability:	20 Set/Sets per Month
Port:	Shanghai
Payment Terms:	L/C,D/A,D/P,T/T,Western Unio

Contact Supplier Chat Now

Start Order Add to Inquiry Cart Add to My

This supplier supports **Trade Assuran**
Follow the Trade Assurance process a

- On-time shipment and pre-shipment produc
- Refund up to the covered amount agreed wi

Friendly Competition aligned with Desired Outcomes

Incentives, even if light handed play a big role

Feature 1 & 5 are prioritized

Wrong order list

Work released to prod
Feature 1
Feature 5
Production Defect
Feature 2
Feature 3
Feature 4
Feature 6

Work released to prod
Feature 1
Feature 5
Production Defect
Feature 2
Feature 3
Feature 4
Feature 6

Thanks to Chris Matts, Lisa Long and John Horton for the “Wrong order’o’meter” concept.

Take-Aways

- Be aware of the impact of dependencies
  - A single dependency reduce order options 50%, < 1% with 4
  - Every dependency removed double your chance of on-time delivery
- Don’t be afraid to have teams up to 15 people
  - if it avoids even a single dependency
- Visualize your dependencies
- Manage your team skill balance to avoid constraints
- Get cookies aligned between teams and dependents

Risk – The Final Enterprise Agile Frontier

- Top 10 reasons forecasting software projects fails
- Not your grandparents risk management
- How and why to do agile risk management

Risk – The Final Enterprise Agile Frontier

10:45 – National Harbor 6/7

Thank You

- Email me: troy.magennis@focusedobjective.com
- Follow me: @t_magennis
- Get spreadsheets and tools: http://Bit.ly/SimResources
- Download the slides: [here]