



Why security initiatives are **doomed** to fail

What you can do about it

Understanding **how work flows** is fundamental

You need to be **constraint-aware**

Agile **doesn't scale** across teams

Developers are both
the cause of and the solution to
all problems

Security is about
influence not execution

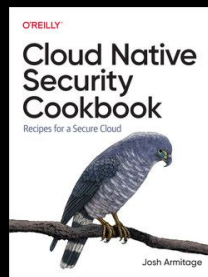
Josh Armitage

Senior Engagement Partner @ Cognizant Servian

 <https://www.linkedin.com/in/josh-armitage-b7825a41/>

 @josharmi

- Worked with some of the world's biggest enterprises
- Reader of books, drinker of wine, supposed knower of things
- Built world-first serverless architectures on AWS
- Has worked with everything from mainframes to machine learning
- International speaker
- O'Reilly author



@josharmi

Quokka Incorporated

Trying to **digitally transform**

"Doing agile"

Security is working harder than ever

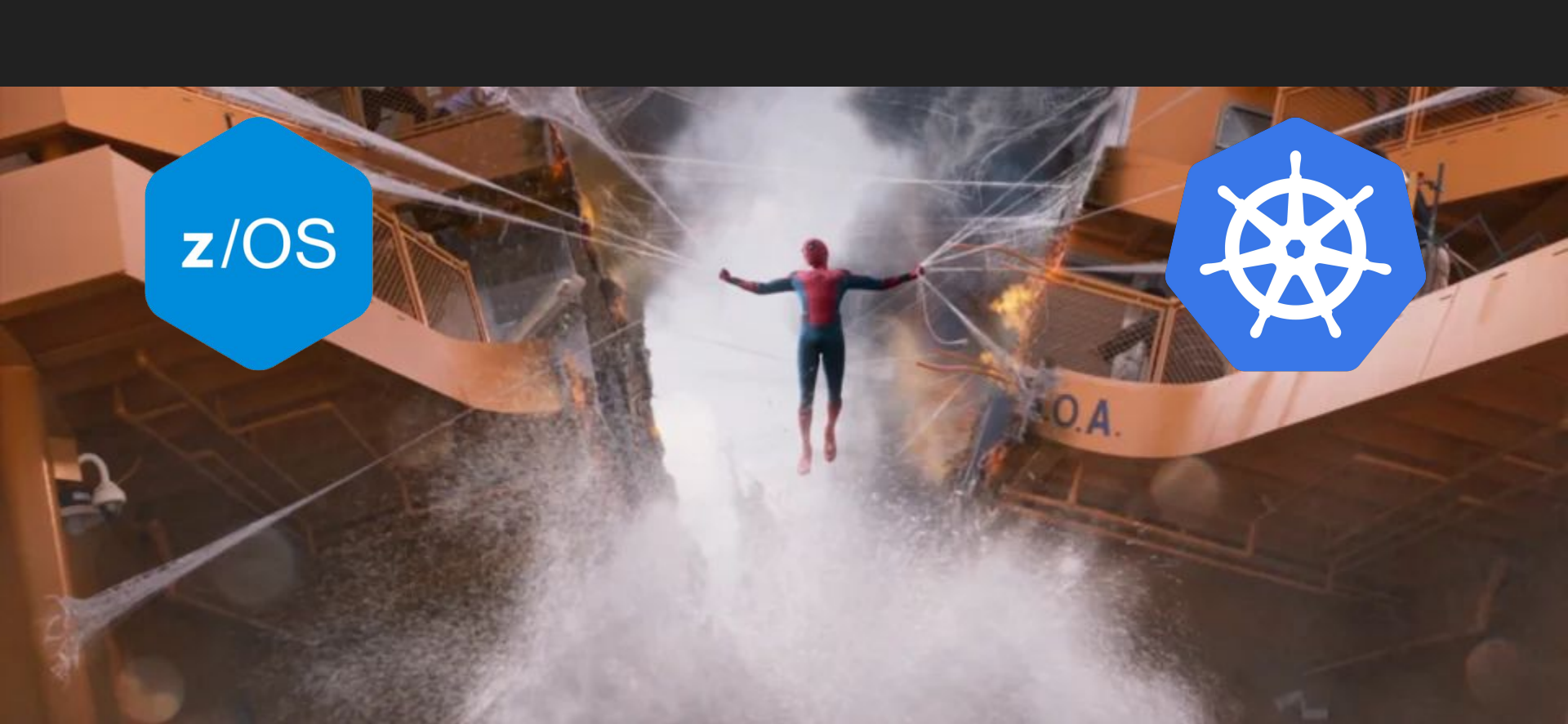
Based on painful real world experiences



@josharmi

Hitting keyboards or people
can only get you so far

Dual Speed IT

A background image of Spider-Man in his red and blue suit, suspended in the air between two orange metal structures of a ship. He is holding onto thin web lines. The scene is filled with a large spray of water or steam, and there are some flames visible on the right side. The ship's structure has some text, including "O.A." on a curved beam.

z/OS



An average day in the life

@josharmi

People > Process > Tools

Try to **tool** their way out of
a people problem

CSPM

Cloud Security Posture Management

~\$1 million per annum

1st year paid by security

Afterwards move to chargeback model

Controls

CSPM controls **increasing 5%** every month

Violations **increasing 10%** month on month

Average age of violation **increasing**

Security Posture

Eroding over time

We're finding problems **faster than we can fix**

Reactive over **proactive** approach

How Work Flows

Value Streams

From “**Please**” to “**Thank You**”

Can be owned by **one team**

Or **split** across them

Value Streams Metrics

Develop

Build

Test

Deploy

Throughput is the units delivered per unit time
=> 1.6 stories per day

Lead Time is the time for a unit to be delivered
=> 3 days

An Organisation

is made of

Value Streams

Value Streams

Product Teams own theirs **end-to-end**

Develop

Build

Test

Deploy

Everything is in **sphere of control**

They **don't get blocked** on other teams

Most **effective way** of delivering customer value

PRODUCT TEAM LESSONS FROM
THE WORLD'S TOP TECH COMPANIES

MARTY CAGAN
Silicon Valley Product Group

INSPIRED

HOW TO
CREATE
TECH
PRODUCTS
CUSTOMERS
LOVE

SECOND EDITION

WILEY

PRODUCT LEADERSHIP LESSONS FROM
THE WORLD'S TOP TECH COMPANIES

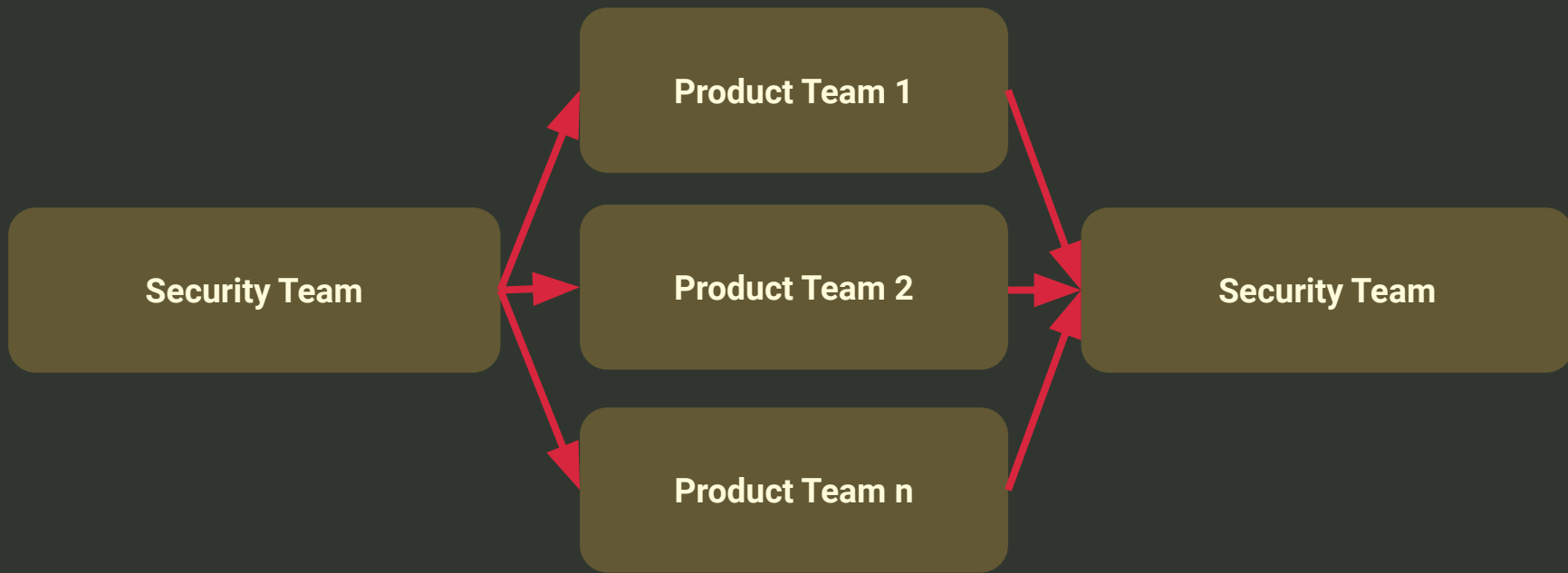
MARTY CAGAN WITH CHRIS JONES
Silicon Valley Product Group

EMPOWERED

ORDINARY
PEOPLE,
EXTRAORDINARY
PRODUCTS

WILEY

Security
isn't a product team



Security streams **feed** into product streams

Becoming Constraint-Aware

Theory of Constraints

In any system there is **exactly one** constraint

The constraint controls **throughput**

Investing away from the constraint is pure **waste**

More project managers

never

makes a project go quicker

1. Find the constraint
2. Optimise the constraint
3. Subordinate to the constraint
4. Elevate the constraint
5. Iterate

Constraints

We want to keep the constraint **optimally** fed

We never want to **starve** the constraint

We never want to **overload** the constraint

Multitasking

Is a **myth**

Task switching and debating priorities are both **wasteful**

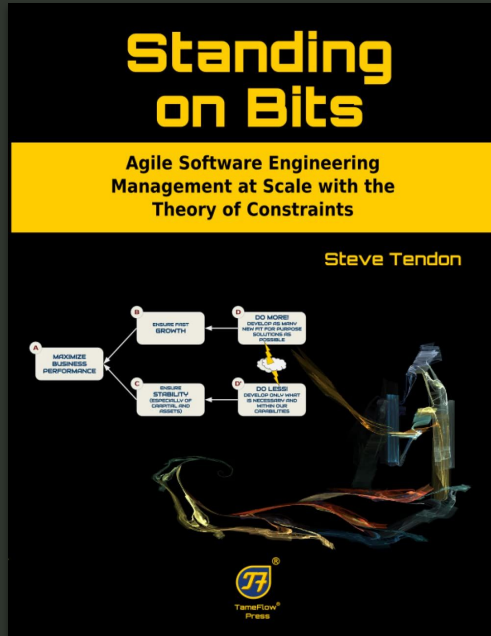
We want to **release work** as the constraint becomes **available**

The constraint is **outside** the security team

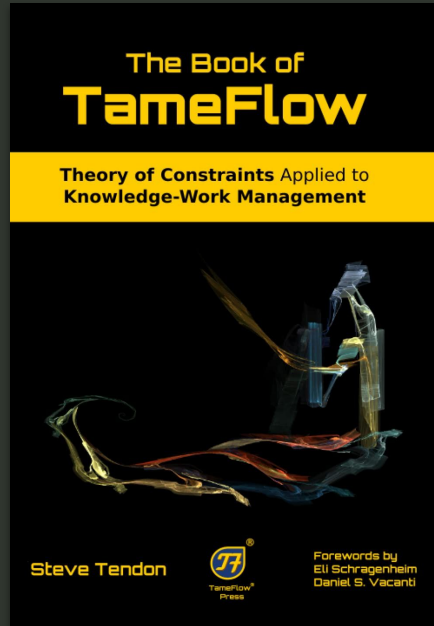
Work enters **sphere of influence**

Need to **understand** and **optimise**

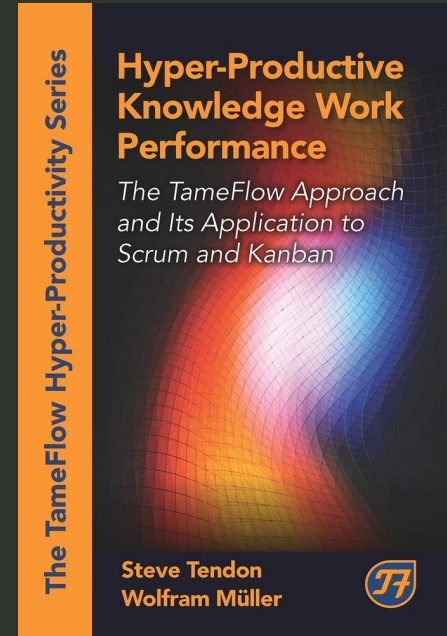
Tameflow



1st



2nd

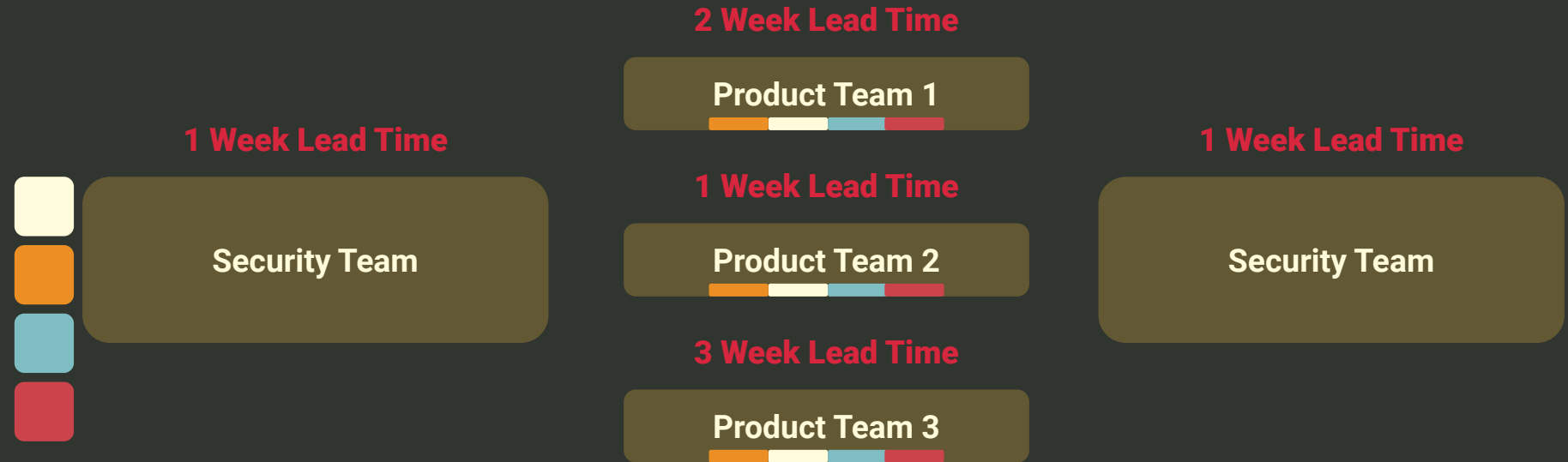


3rd

Infrastructure Configuration Change



Infrastructure Configuration Change



After 1 Week

2 Week Lead Time



Product Team 1



1 Week Lead Time

1 Week Lead Time



Product Team 2



3 Week Lead Time



Product Team 3

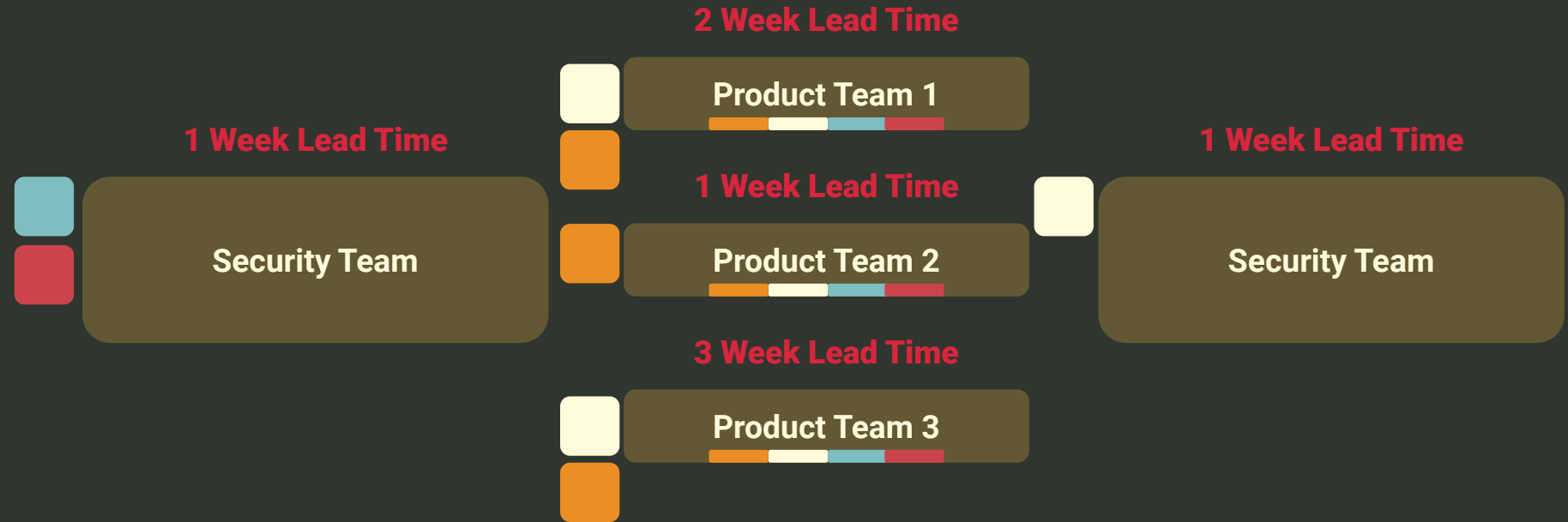


1 Week Lead Time

Security Team

Security Team

After 2 Weeks



After 3 Weeks

2 Week Lead Time

1 Week Lead Time

1 Week Lead Time

1 Week Lead Time

3 Week Lead Time

Product Team 1

Product Team 2

Product Team 3

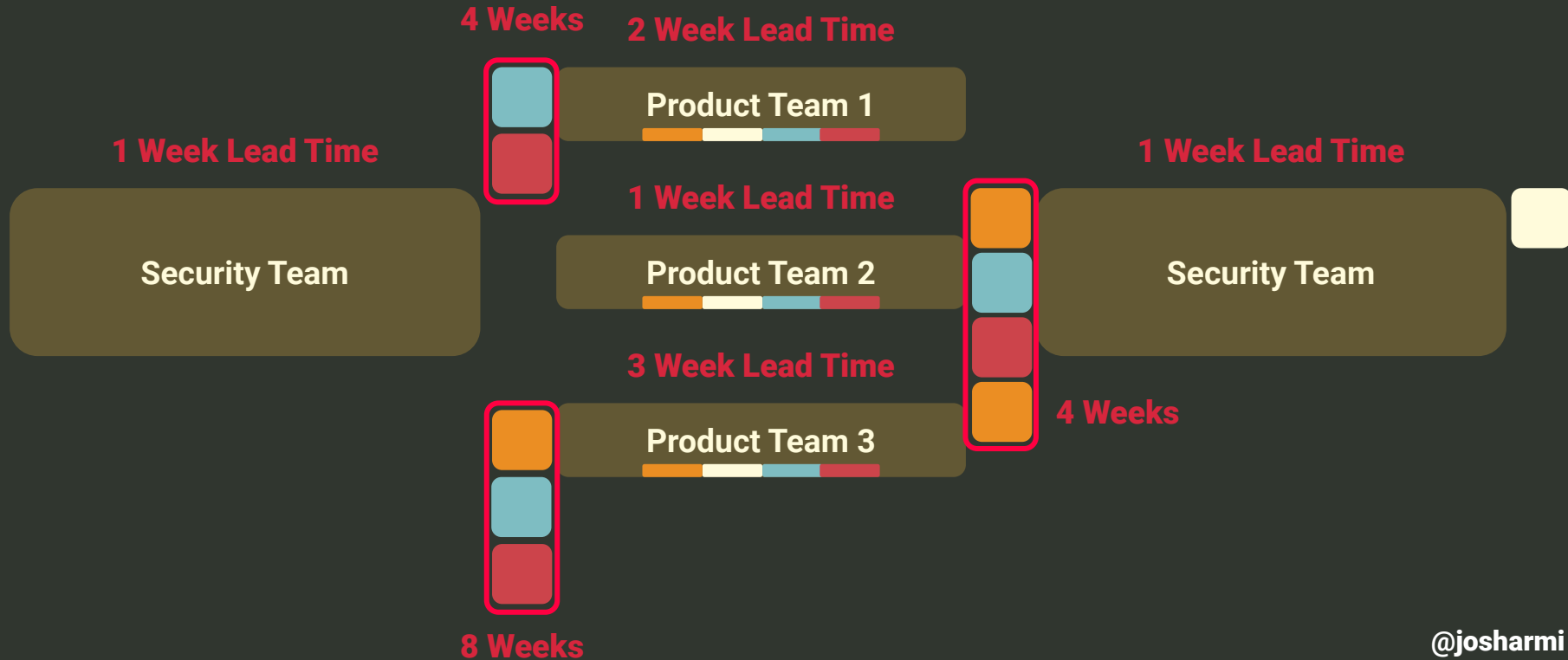
Security Team

Security Team

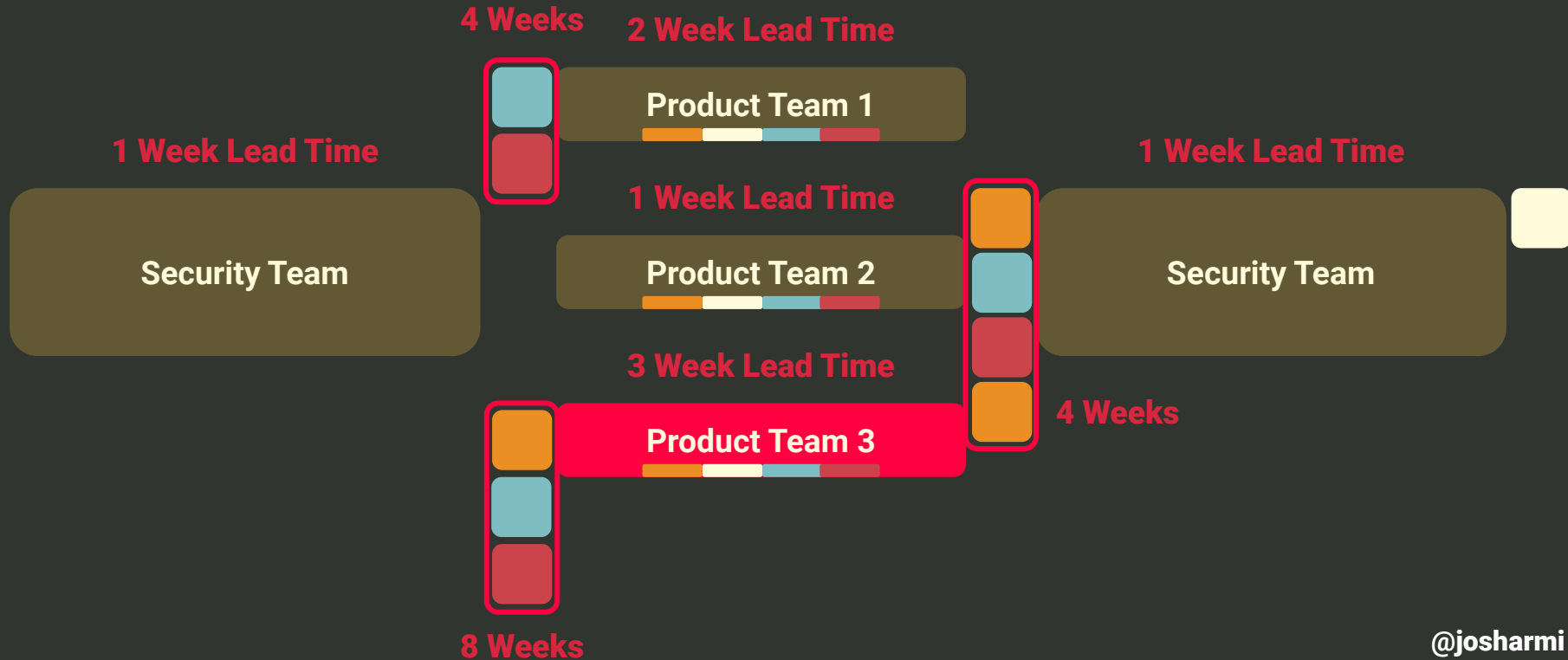
After 4 Weeks



After 5 Weeks



After 5 Weeks

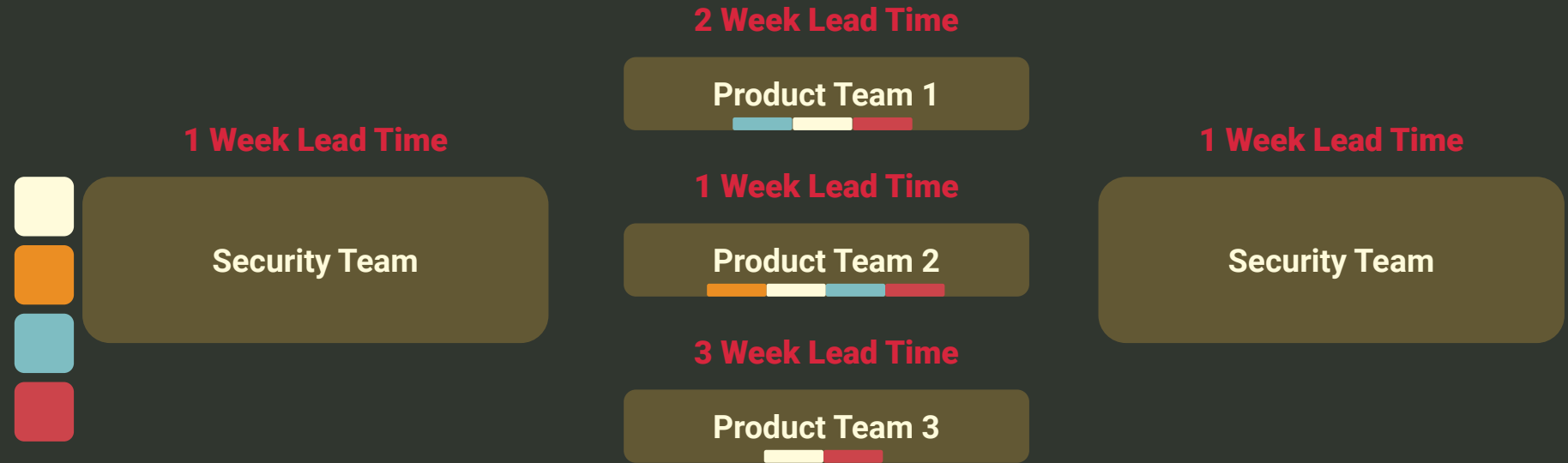


Finding the Constraint

For a given backlog of work

The team with the longest queue in **time**

Infrastructure Configuration Change



After 1 Week

2 Week Lead Time



Product Team 1



1 Week Lead Time

1 Week Lead Time



Product Team 2



Security Team

3 Week Lead Time



Product Team 3



1 Week Lead Time

Security Team

After 2 Weeks



After 3 Weeks

2 Week Lead Time



1 Week Lead Time



1 Week Lead Time



1 Week Lead Time



3 Week Lead Time



After 4 Weeks



After 5 Weeks

1 Week Lead Time

Security Team

2 Week Lead Time

Product Team 1

1 Week Lead Time

Product Team 2

3 Week Lead Time

Product Team 3

1 Week Lead Time

Security Team

After 5 Weeks



After 12 weeks

Constraint Naive

Throughput: 3 Items

Lead Time: 4 Weeks

Constraint Aware

Throughput: 7 Items

Lead Time: 1.7 Weeks

All we did was
shape the work
around the constraint

Working **smarter**
not **harder**

Security in an Agile World

Agile Planning



Professional Fortune Telling



WSJF

Weighted Shortest Job First

Cost of Delay

Job Size

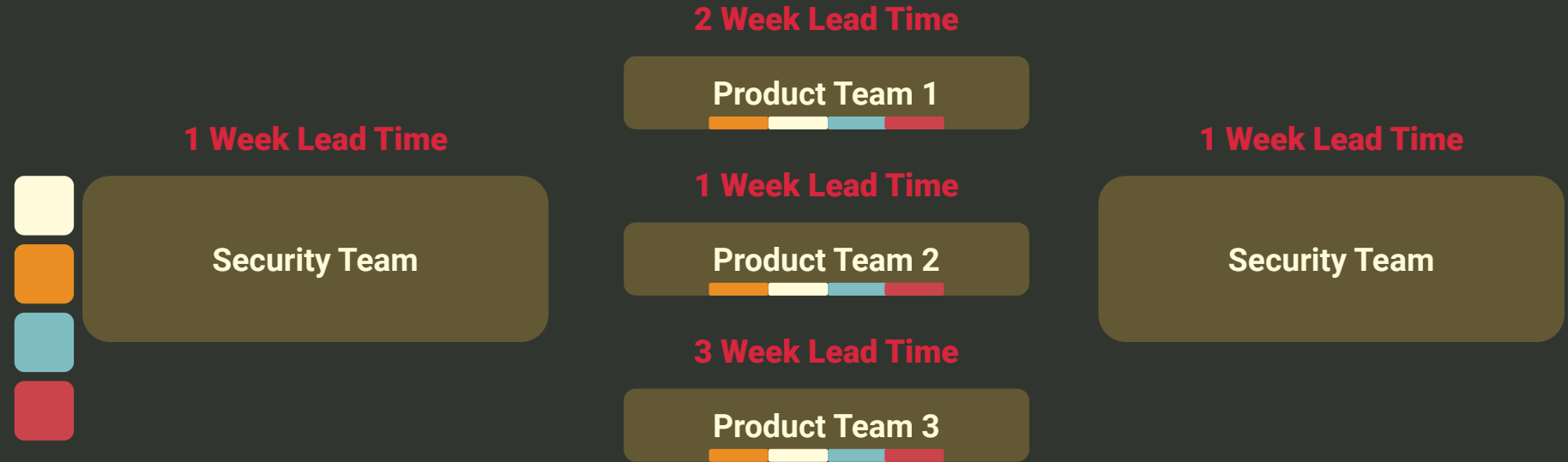
Value

Effort

WSJF in **Multi-Team** Scenarios

Job Size

Constraint Unaware



Constraint Aware

2 Week Lead Time

Product Team 1



1 Week Lead Time

Product Team 2



3 Week Lead Time

Product Team 3



1 Week Lead Time

Security Team

1 Week Lead Time

Security Team

Has job size
changed?

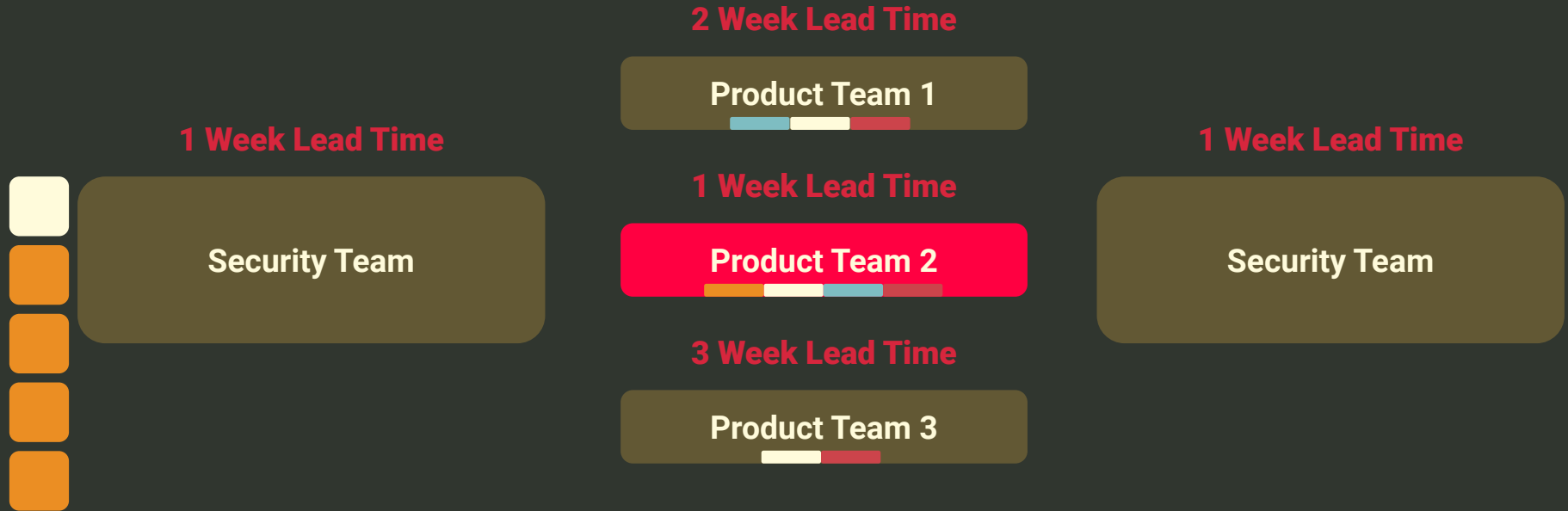
Effort is
impact at the constraint

The constraint is
outside the team

Maximising value

shaping the work to the system

Work Shape Dictates Constraint



What about story points?

50 points

Security Team

35 points

Product Team 1

80 points

Product Team 2

25 points

Product Team 3

50 points

Security Team

50 points every 2 weeks

Security Team

35 points every 2 weeks

Product Team 1

80 points every 4 weeks

Product Team 2

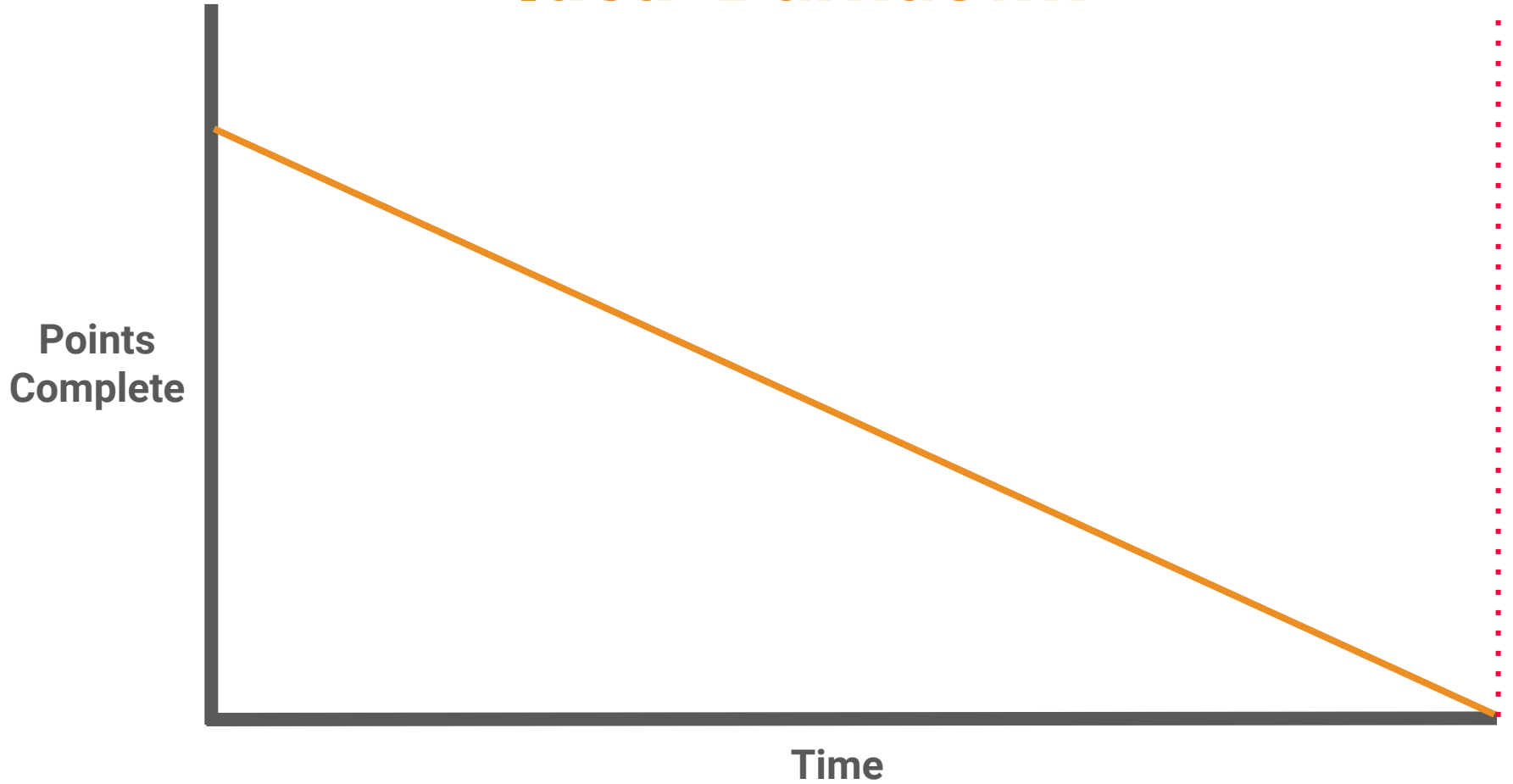
25 points every 2 weeks

Product Team 3

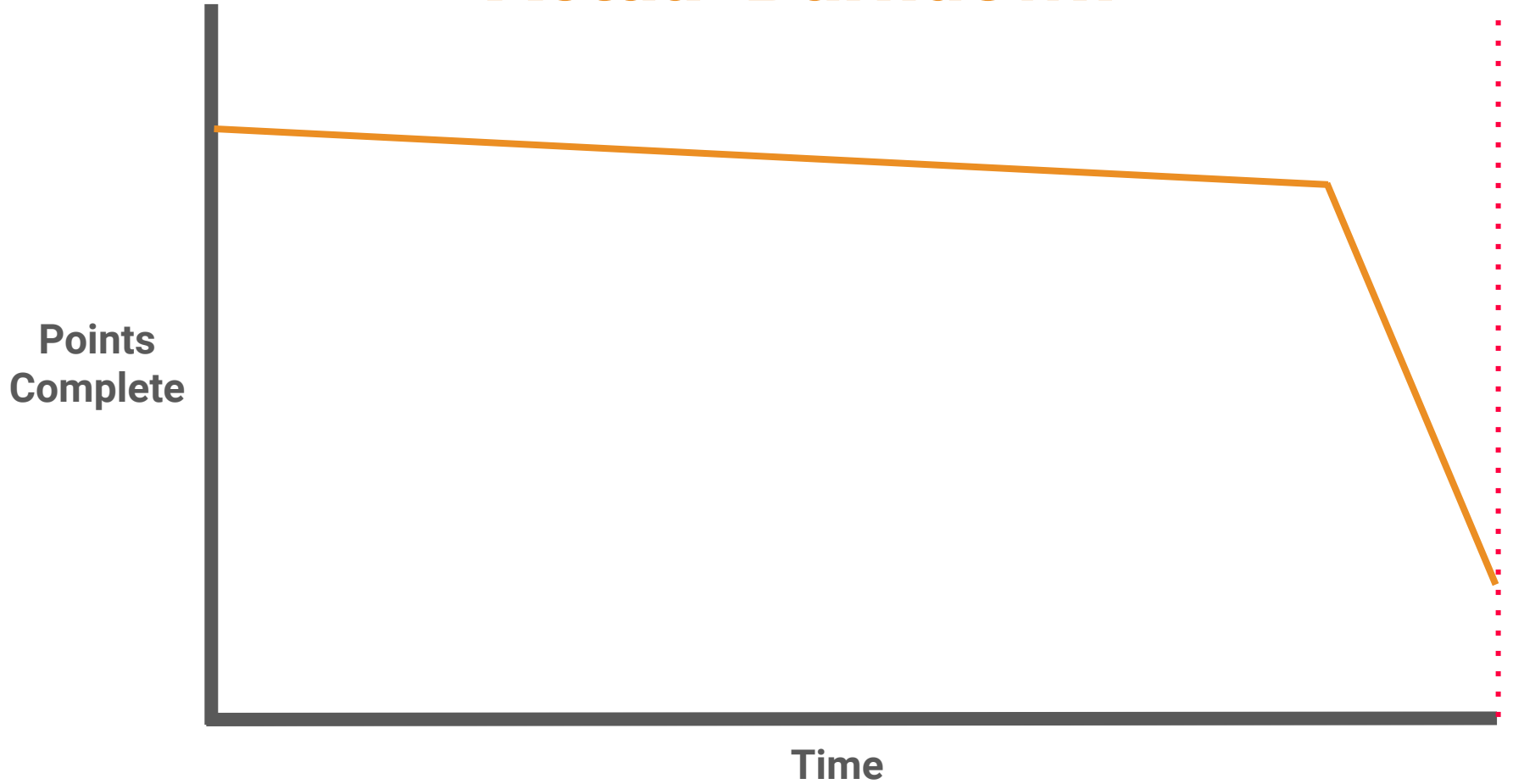
50 points every 2 weeks

Security Team

Ideal Burndown



Actual Burndown



Chasing Predictability

You need a **stable** system

Stable

2 Week Lead Time

Product Team 1



1 Week Lead Time

Product Team 2



3 Week Lead Time

Product Team 3



1 Week Lead Time

Security Team

1 Week Lead Time

Security Team

Unstable

2-5 Week Lead Time

Product Team 1



1-3 Week Lead Time

Product Team 2



3-9 Week Lead Time

Product Team 3

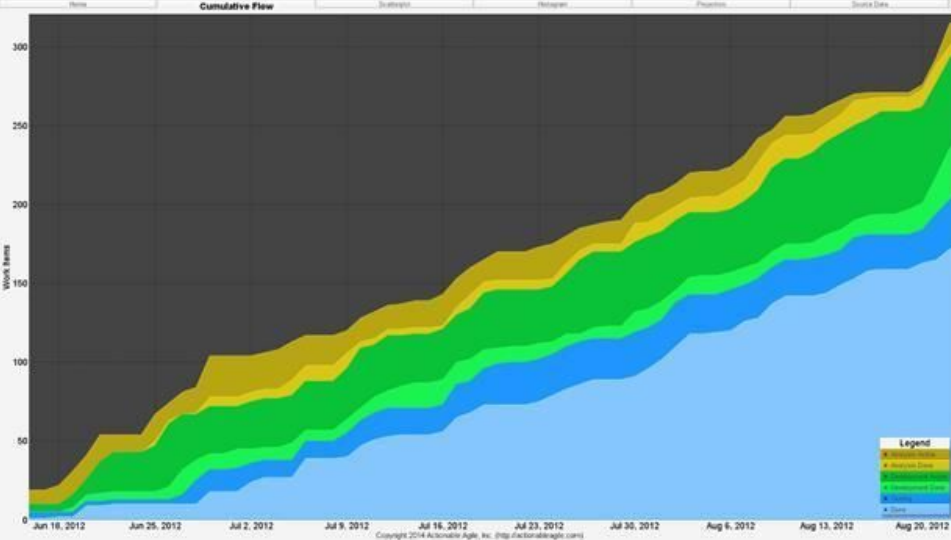


1 Week Lead Time

Security Team

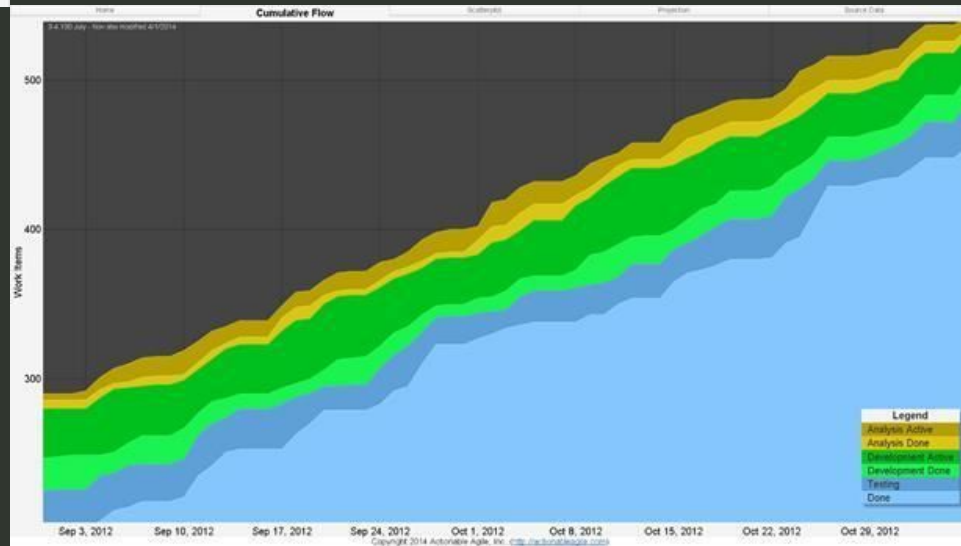
1 Week Lead Time

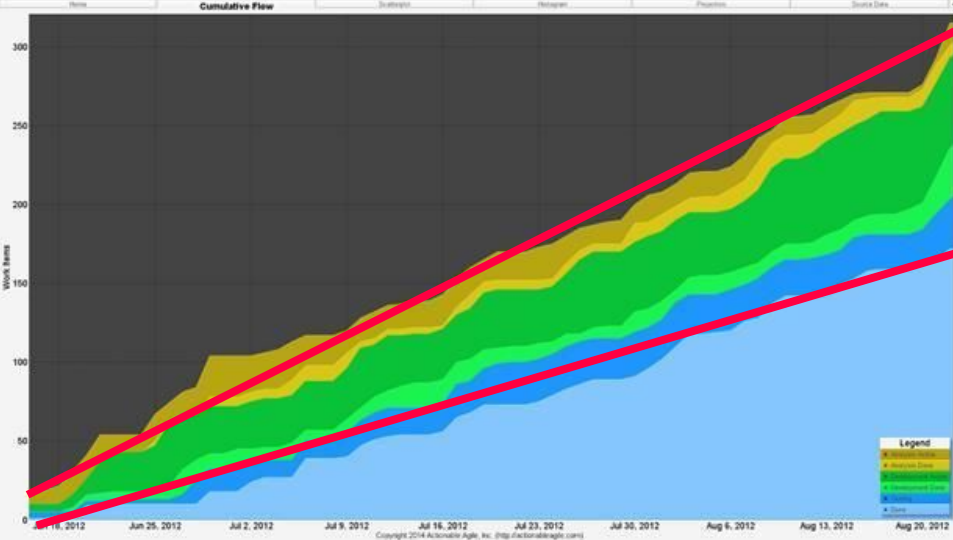
Security Team



<= Less Like This

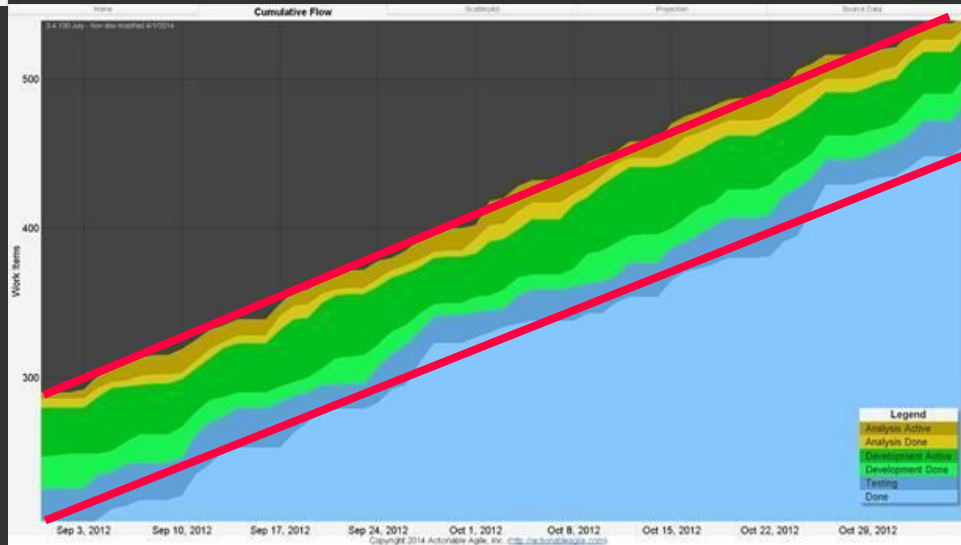
More Like This =>



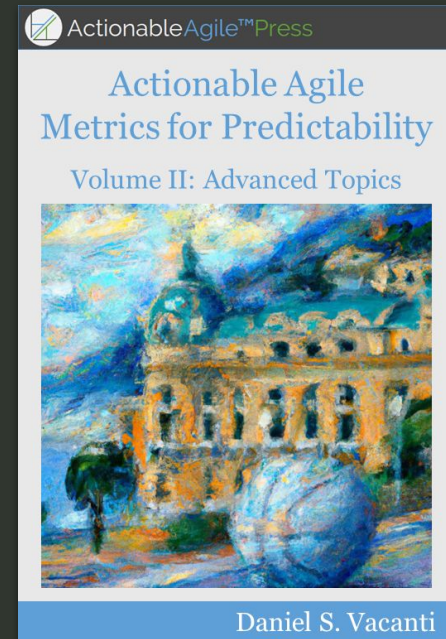
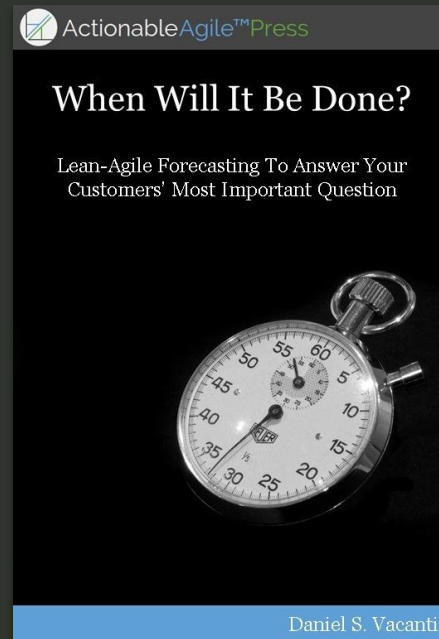
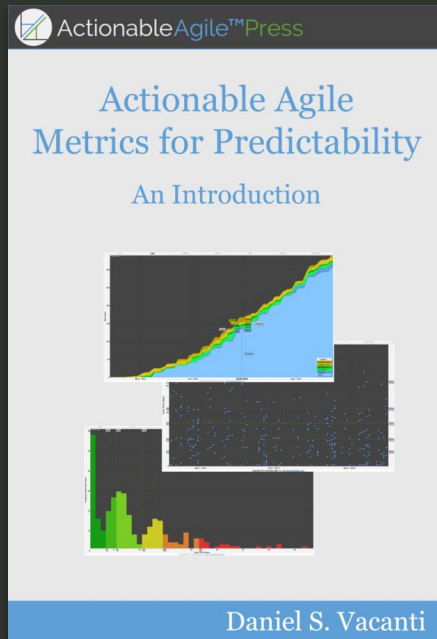


<= Less Like This

More Like This =>



Actionable Agile Metrics



Value Stream Mapping

"The new bible for value stream mapping and improving organizational performance."

—Art Byrne, former CEO, The Wiremold Company, and author, *The Lean Turnaround*

**VALUE
STREAM
MAPPING**



How to
Visualize Work and
Align Leadership
for Organizational
Transformation

Karen Martin

Author of the Shingo Research Award-winning *The Outstanding Organization*

AND Mike Osterling

The Addison-Wesley Signature Series

A MARTIN FOWLER SIGNATURE
BOOK
Martin

CONTINUOUS DELIVERY

RELIABLE SOFTWARE RELEASES THROUGH BUILD,
TEST, AND DEPLOYMENT AUTOMATION

JEZ HUMBLE
DAVID FARLEY



Foreword by Martin Fowler

Continuous Delivery

@josharmi

Protecting Capacity for Security

Security work gets
constantly deprioritised

Flow Distribution



Feature

Risk

Defect

Debt

What **one decision** can I make
that prevents the
next one hundred?

Benevolent Dictatorship Prioritisation

Feature
Product Owner

Risk
Security

Defect
Customer

Debt
Team

Hyperbolic Discounting

Hyperbolic Discounting

Short-term gain for long-term pain

Feature
Product Owner

Risk
Security
20%

Defect
Customer

Debt
Team
20%

Feature
Product Owner

First
Risk
Security
20%

Defect
Customer

First
Debt
Team
20%

Stable

2 Week Lead Time

Product Team 1



1 Week Lead Time

Product Team 2



3 Week Lead Time

Product Team 3



1 Week Lead Time

Security Team

1 Week Lead Time

Security Team

Where Are We Now?

We've maximised the
system we have

1. Find the constraint
2. Optimise the constraint
3. Subordinate to the constraint
4. Elevate the constraint
5. Iterate

1. Find the constraint
2. Optimise the constraint
3. Subordinate to the constraint
- 4. Elevate the constraint**
5. Iterate

Shift Left

Push Down

Move Empowerment

Shift Left

Push Down

Move Empowerment

Google's Large Scale Change Approach

Removes multi-team value streams

Google's Large Scale Change Approach

Removes multi-team value streams

Accountability for service belongs to product teams

Google's Large Scale Change Approach

Removes multi-team value streams

Accountability for service belongs to product teams

Empower security teams to make changes **autonomously**

Log4j in a single change

Google's Large Scale Change Approach

Removes multi-team value streams

Accountability for service belongs to product teams

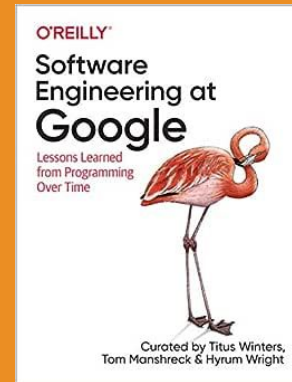
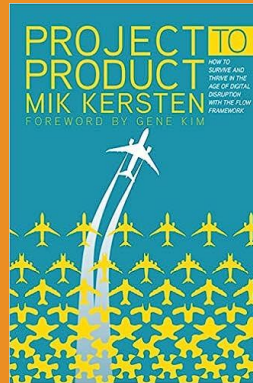
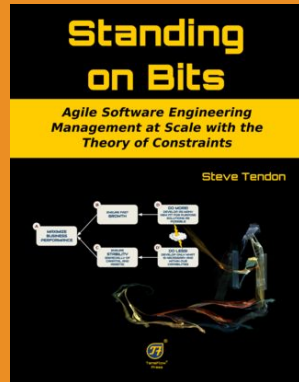
Empower security teams to make changes **autonomously**

Requires **world-leading** development & operations

Understanding **how work flows** is fundamental

You need to be **constraint-aware**

Agile **doesn't scale** across teams



Agile Release Trains
are just SAFe's version of the
trolley problem

Understanding **how work flows** is fundamental

You need to be **constraint-aware**

Agile **doesn't scale** across teams

