DevOps Mindset & DevOps Metrics

Benjamin Day
TRAINER | COACH | DEVELOPER

@benday   www.benday.com

Summary

DevOps mindset

What is work?

DevOps metrics

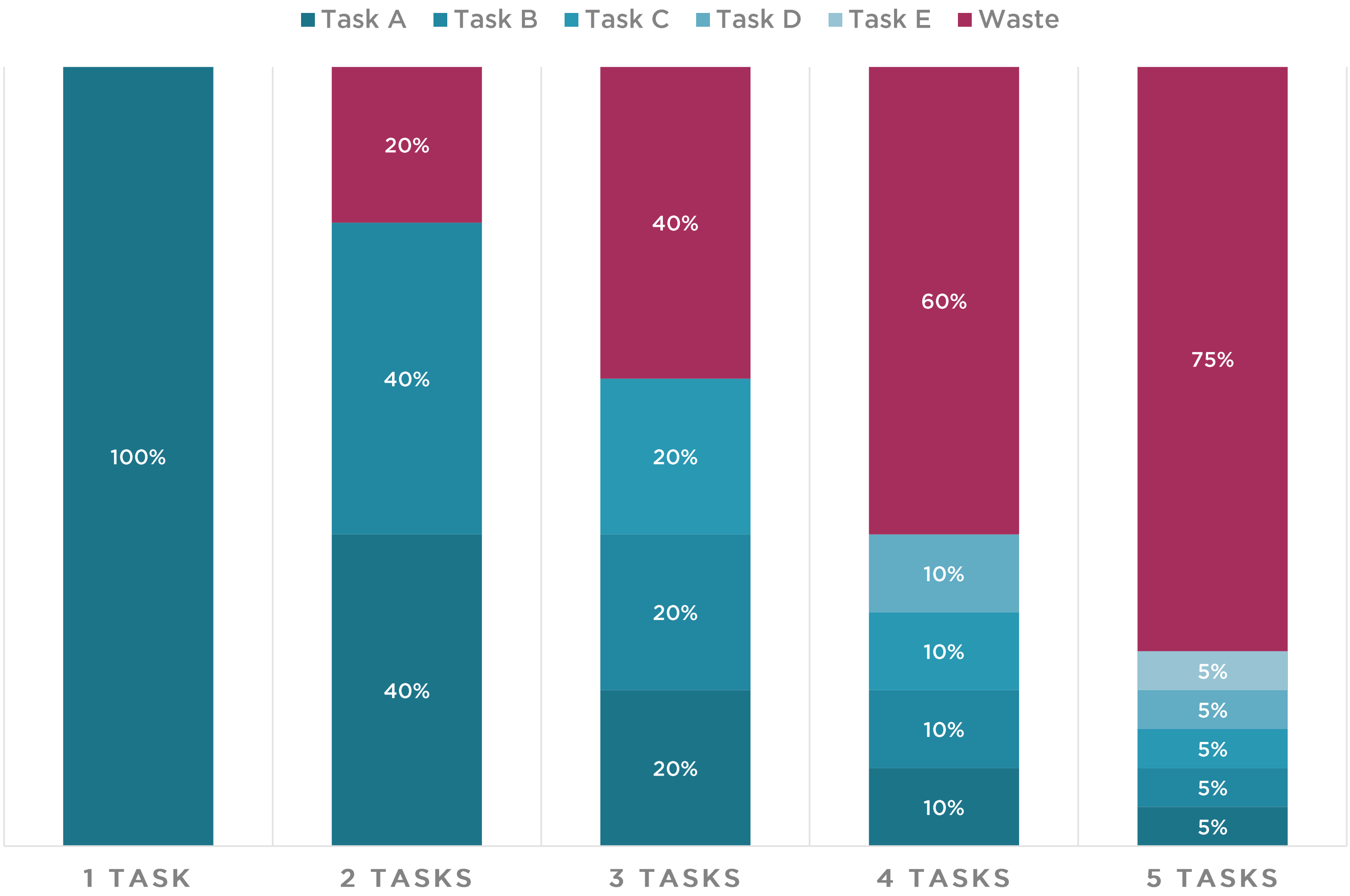
Theory of Constraints

Next up:
Work

How’s life going?

Productivity vs. Waste

0% waste

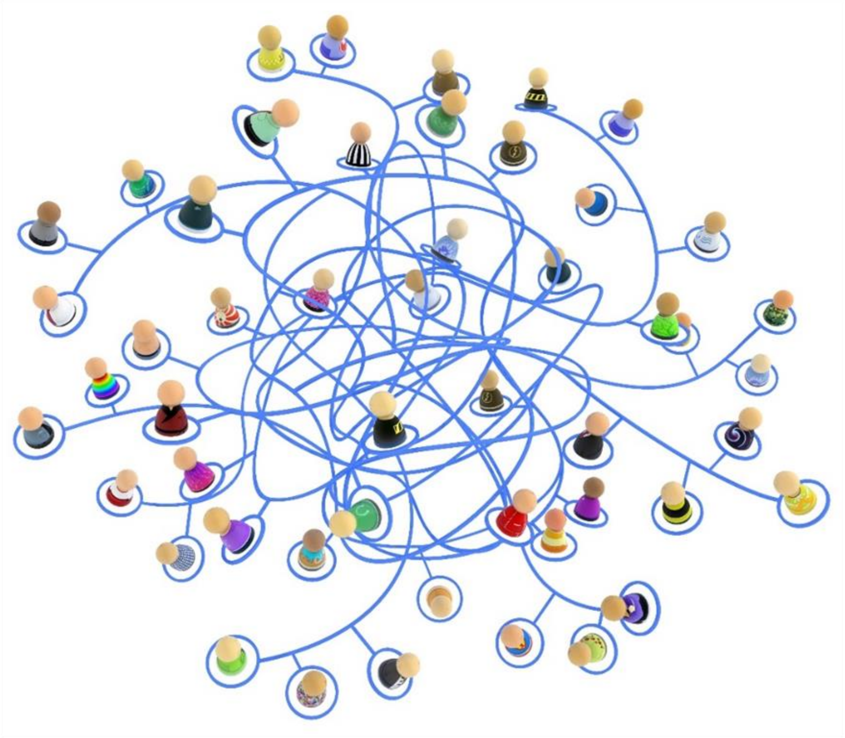


75% waste

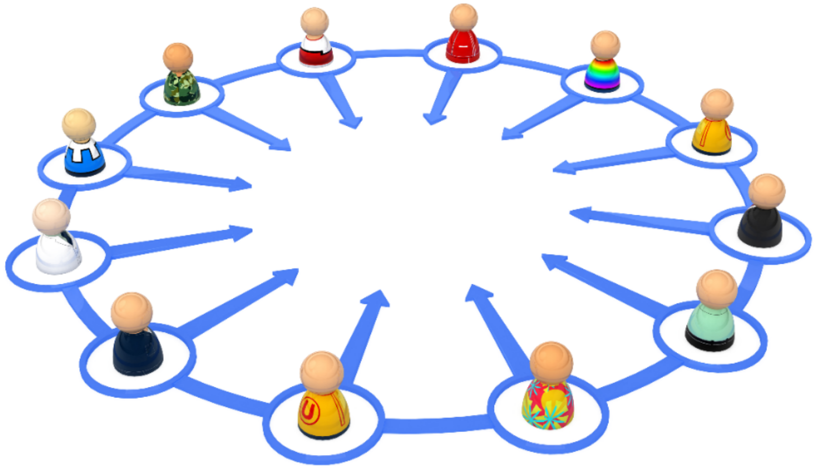
"Quality Software Management: Vol. 1 System Thinking", Gerald Weinberg (1992)

The busier you get,
the more you have to worry.

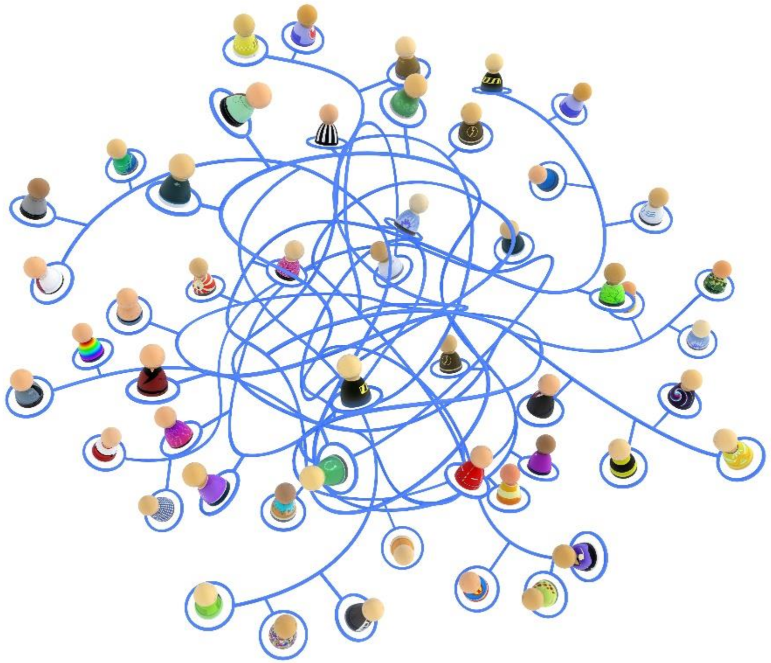
How Do You Feel?



Busy?



Focused?



Are you busy?

Busy is not necessarily good

You want…

- Sustainable pace
- No heroics

Productive teams

No burnout

More on Burnout

Scrum Master Skills

by Benjamin Day

This course will help you develop the skills of a great Scrum Master.

“Dysfunction” module

https://www.pluralsight.com/courses/scrum-master-skills

TFS can help.

Automate tedious & repetitive work

TFS can help…
…but it can’t fix everything.

The people should lead and TFS should follow.

Do less at once
to achieve more overall.

A lot of your work is hiding in plain view.

Where does your work come from?

Four Types of Work in IT

Business Projects

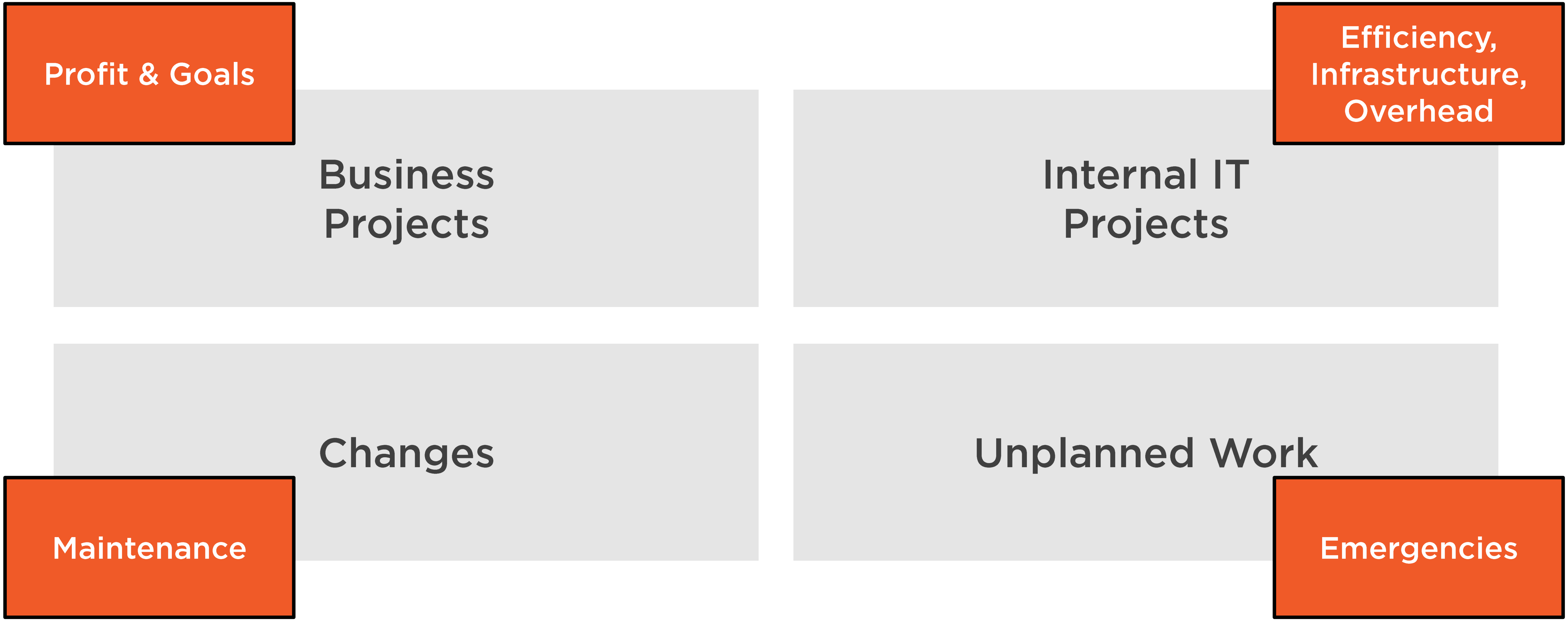
Internal IT Projects

Changes

Unplanned Work

"The Phoenix Project" by Gene Kim

Four Types of Work in IT



Profit & Goals

Business Projects

Efficiency, Infrastructure, Overhead

Internal IT Projects

Changes

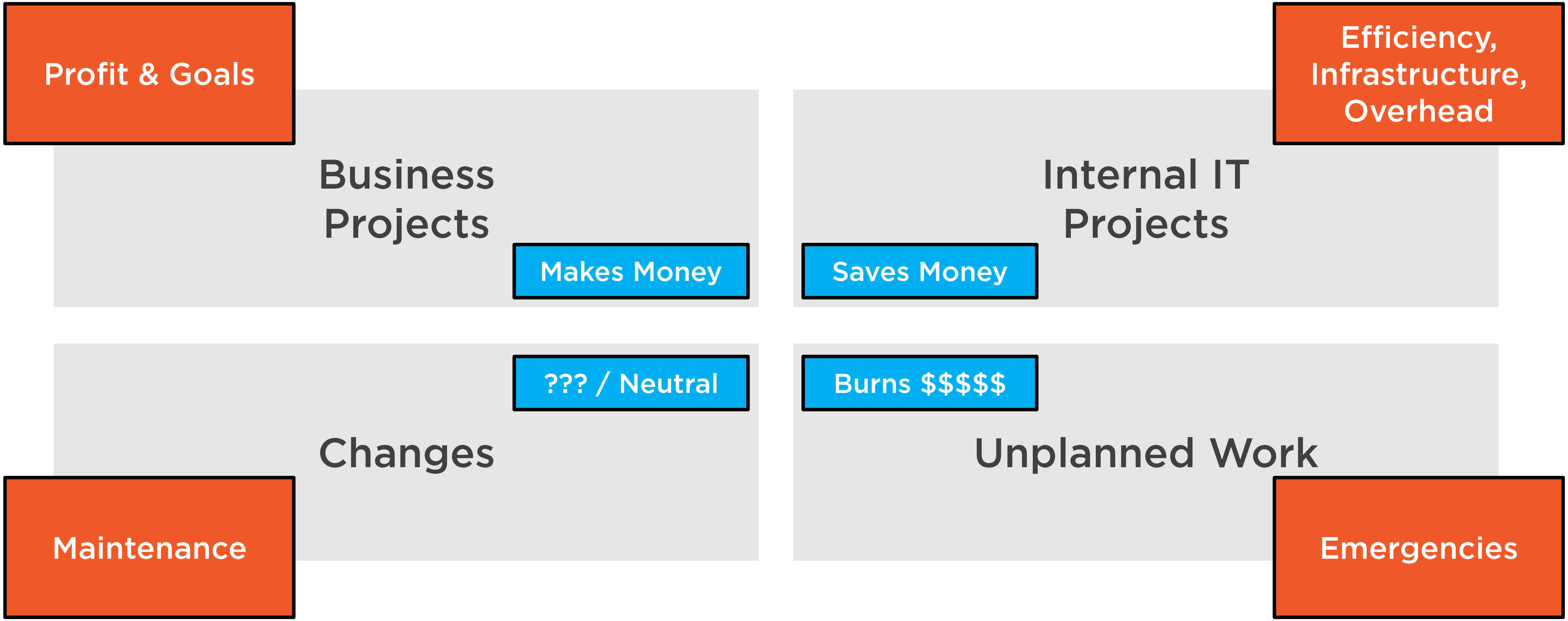
Maintenance

Unplanned Work

Emergencies

"The Phoenix Project" by Gene Kim

Four Types of Work in IT



Profit & Goals

Business Projects

Makes Money

Efficiency, Infrastructure, Overhead

Internal IT Projects

Saves Money

??? / Neutral

Changes

Maintenance

Burns $$$$$

Unplanned Work

Emergencies

“The Phoenix Project” by Gene Kim

Four Types of IT Work

The danger is when…

- …work isn't tracked
- …work isn't tracked in the same way

Hidden work robs you of focus

Unplanned work is especially evil

- "Drop everything and fix it!"

Remember: It's all just work!

You need to pay attention to and track what’s happening.

Why Pay Attention?

Process mindfulness

Notice what you’re working on

Notice where it’s coming from

Notice what kind of work it is

Question why you’re doing it

Noticing helps you get more done

Noticing helps you improve

Do less at once
to achieve more overall.

Next up:
DevOps Metrics & The Theory of Constraints

DevOps Metrics

Are we getting better at DevOps?

What are our DevOps problems?

How are we doing?

DevOps Effectiveness Questions

How long does it take to go from “ask” to “delivery”?

How long does it take a team to deliver once they start working?

How often do you have to work on a production defect?

What % of your time do you work on production defects?

How often to you deploy?

How often does a deployment fail or get rejected?

“How are we doing?”

Lead Time

Cycle Time

“How are we doing?”

Lead Time

- From: Requirement created
- To: Requirement delivered

Cycle Time

- From: Team starts a requirement
- To: Requirement delivered

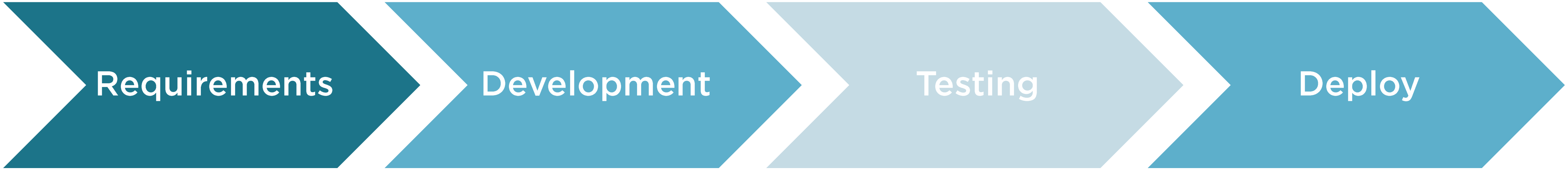
“How are we doing?”

Lead Time	Cycle Time
What does delivery look like to your customers & stakeholders?	How long does it take your team to develop & deliver features?
How fast can you ship a feature or bug fix?	What can you team actually achieve?

“Why is work piling up? Why are we slow?”

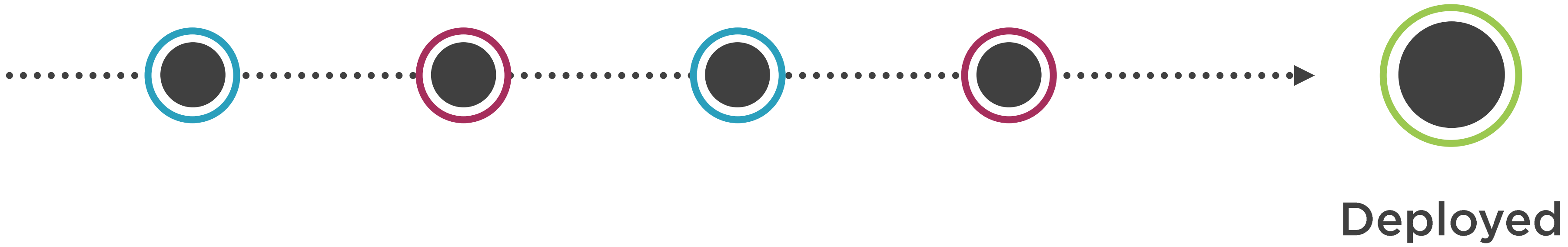
~Measures arrival rate	~Measures completion rate
“We come up with a zillion new ideas.”	“The team is slow.”

Software Delivery Is a Process

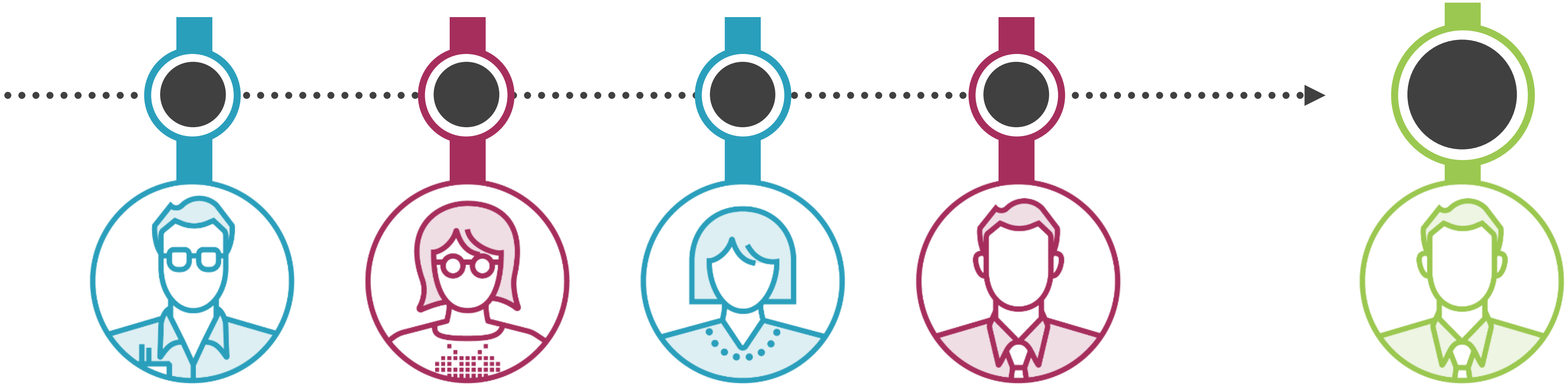


Disclaimer: this is not an endorsement of Waterfall-based project management

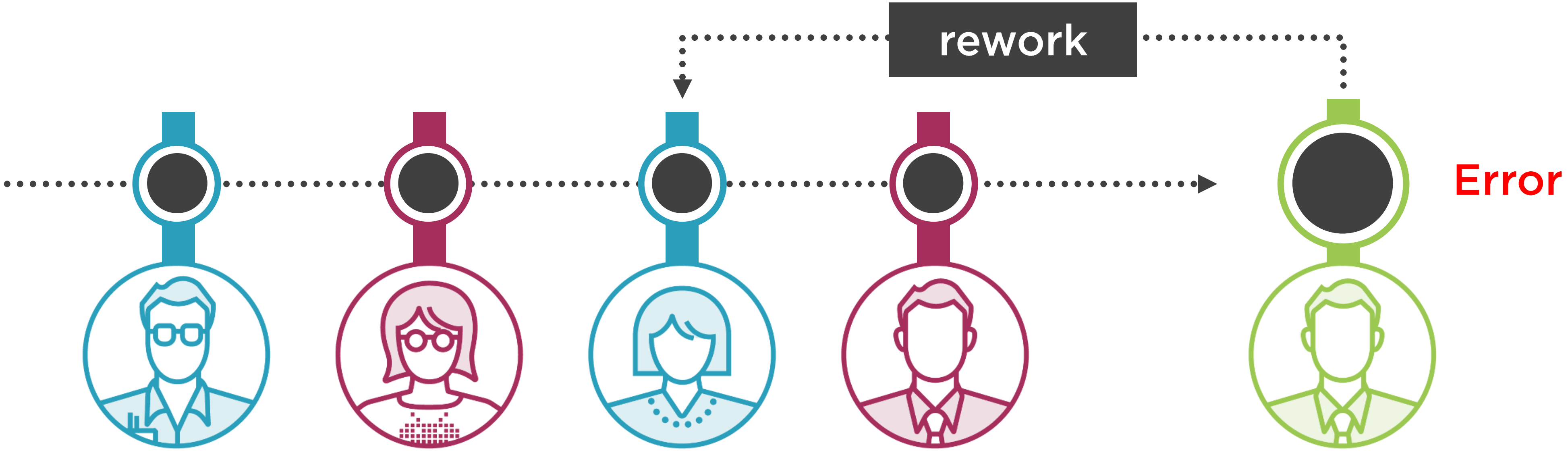
DevOps Is a Flow



DevOps Is a Flow



DevOps Is a Flow



Rework

Anything that has to go backwards in your DevOps flow

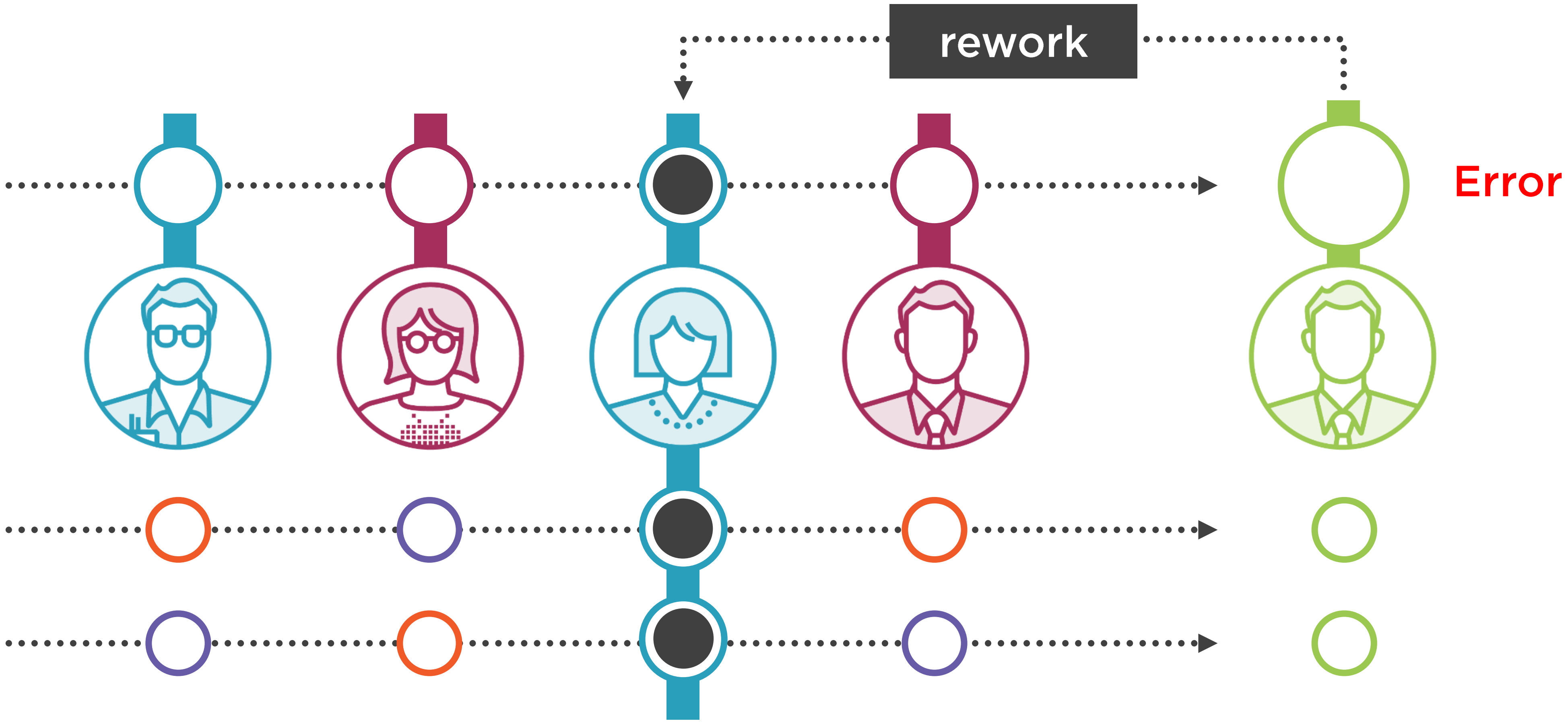
Disruptions

Waste

Slows you down

Fix this → Go faster

DevOps Is a Flow



Wait time

% Busy Time / % Idle Time

Wait time

% Busy	% Idle	Wait Time
20	80	0.25
50	50	1
80	20	4
90	10	9
99	1	99

“90 / 10” is 9x busier than “50 / 50”

“99 / 10” is 11x busier than “90 / 10”

Rework is more painful for busy resources

Are we blocking on someone or something that’s busy?

Constraints

“Bottlenecks”

Introduced by Dr. Eliyahu Goldratt in “The Goal: A Process of Ongoing Improvement”

Theory of Constraints

If you want to go faster, optimizing constraints is the only thing that matters.

Goldratt's Five Steps for Constraints

Identify the constraint

Exploit the constraint

- Maximize the constraint

Subordinate all other activities to the constraint

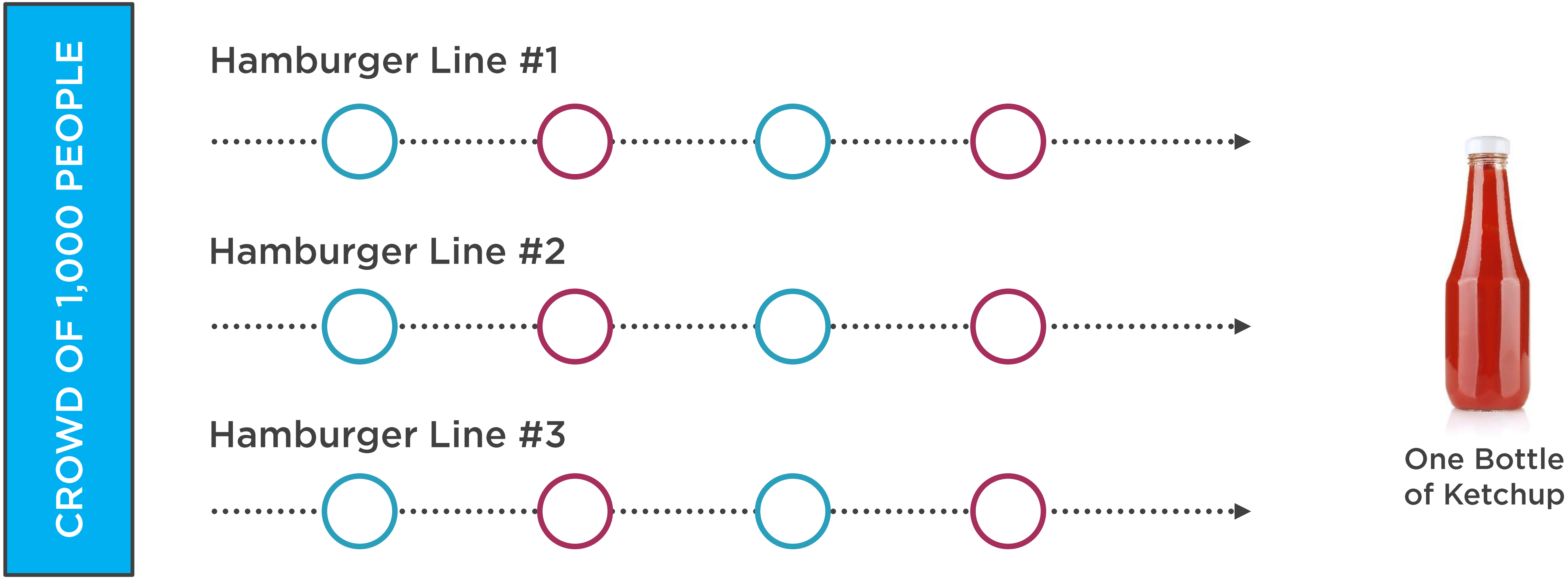
- Avoid producing more than the constraint can handle

Elevate the constraint to new levels

- Expand the capacity of the constraint

Find the next constraint

Constraints in the Lunch Line



CROWD OF 1,000 PEOPLE

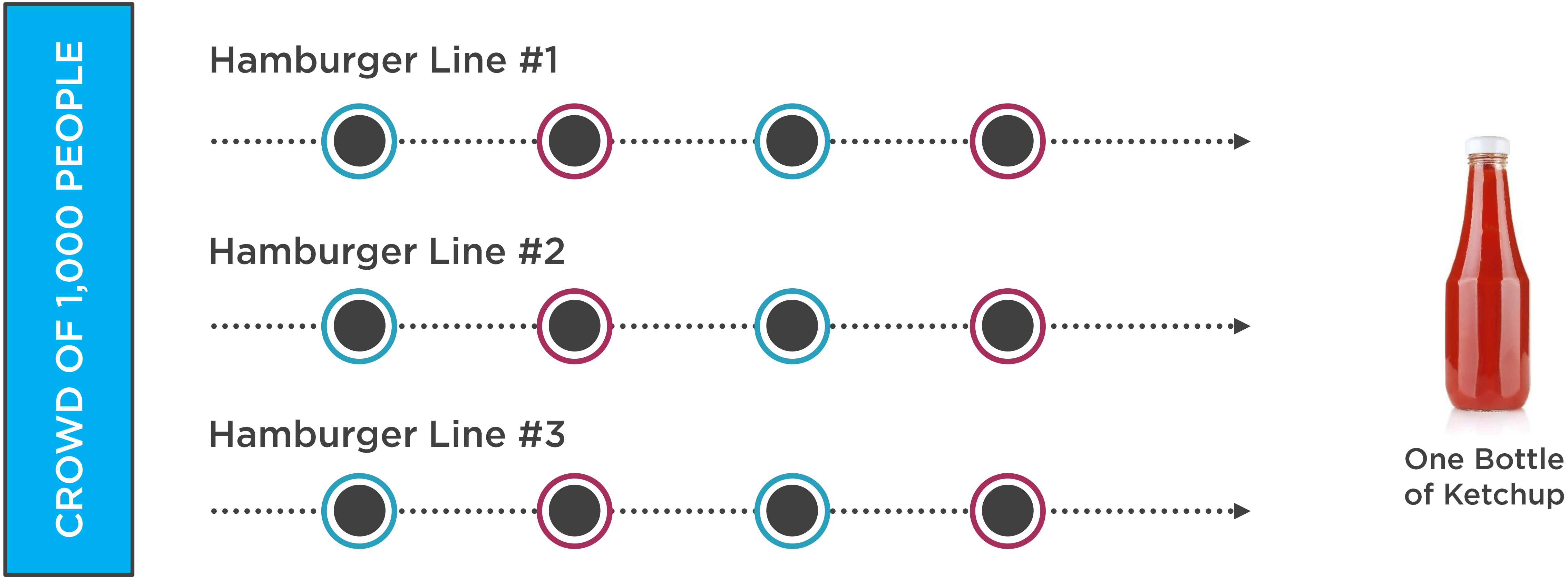
Hamburger Line #1

Hamburger Line #2

Hamburger Line #3

One Bottle
of Ketchup

Constraints in the Lunch Line



CROWD OF 1,000 PEOPLE

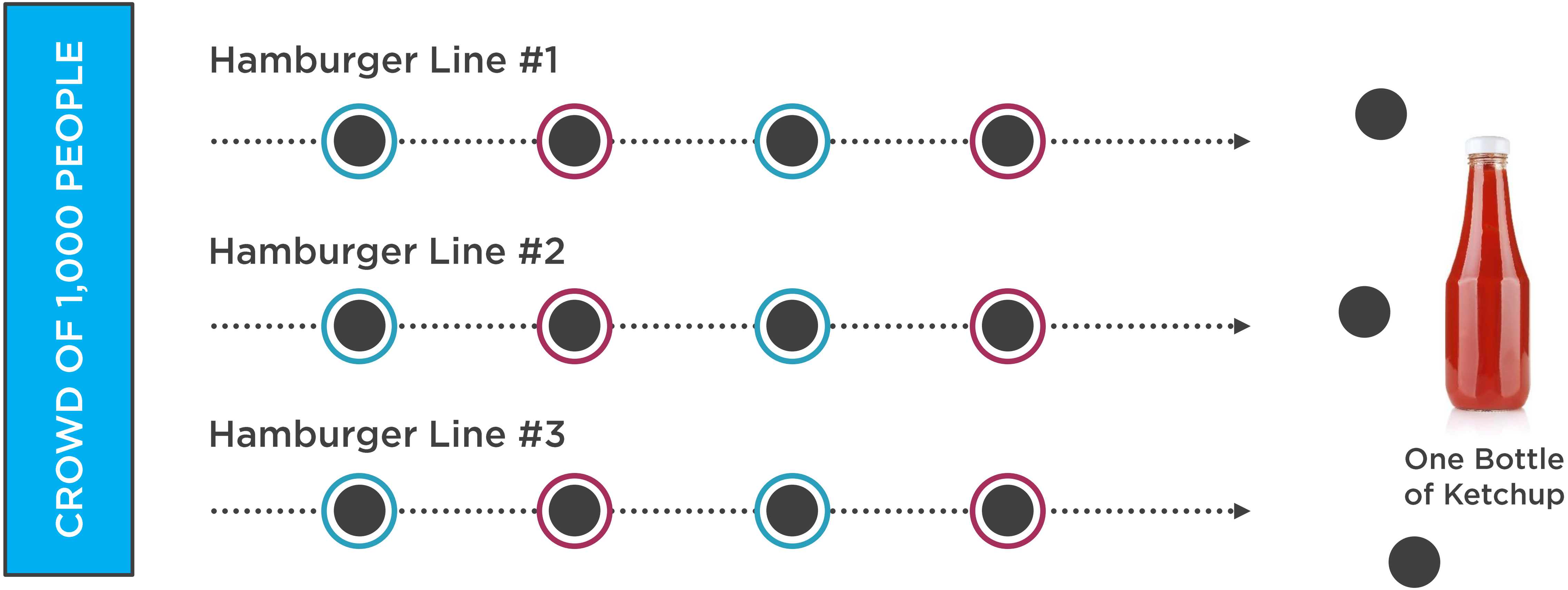
Hamburger Line #1

Hamburger Line #2

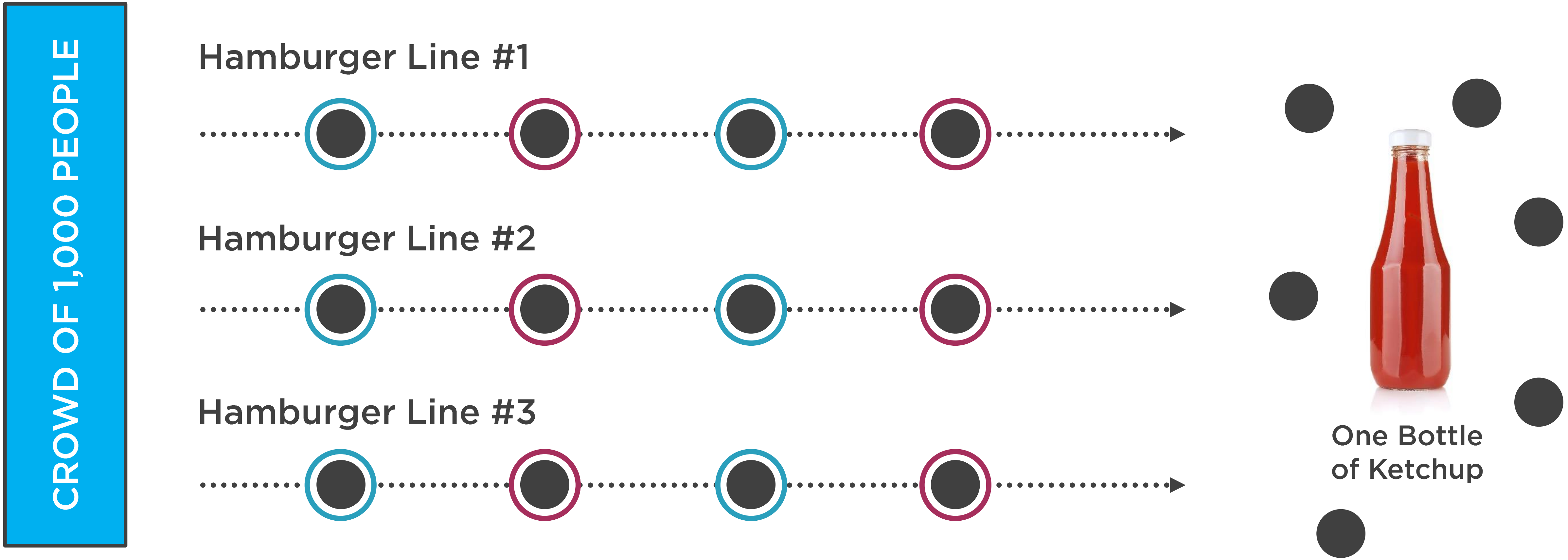
Hamburger Line #3

One Bottle
of Ketchup

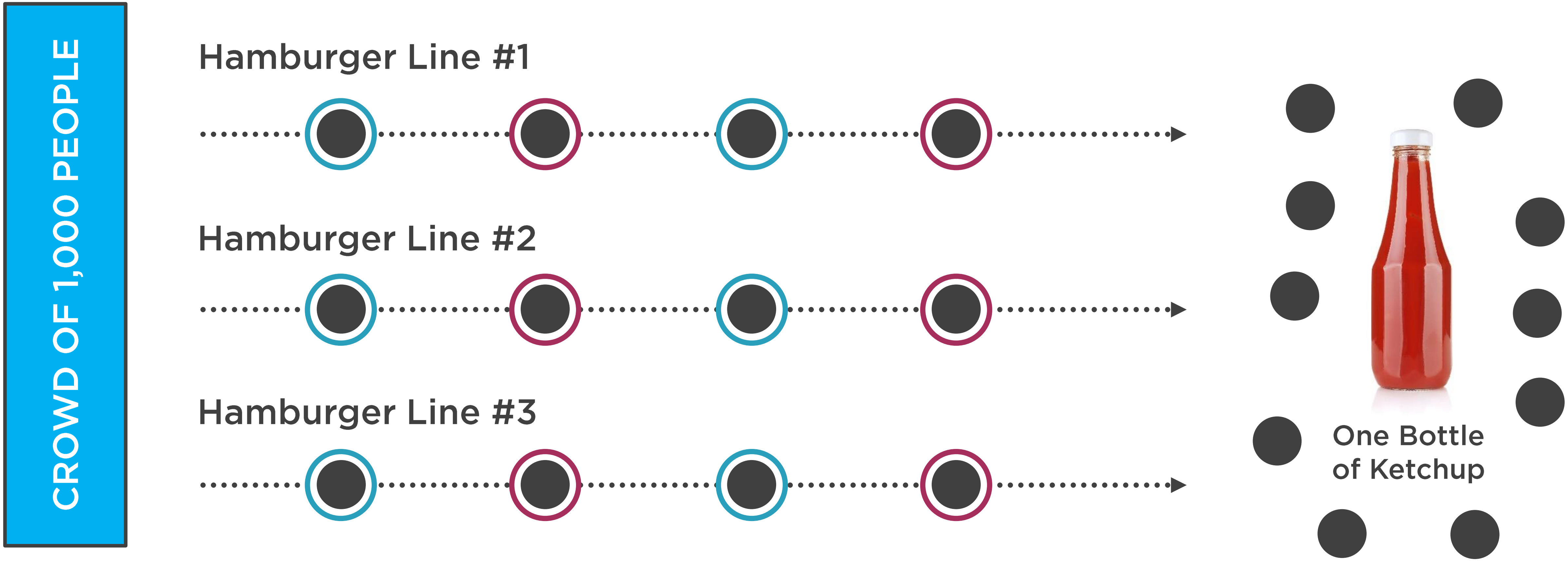
Constraints in the Lunch Line



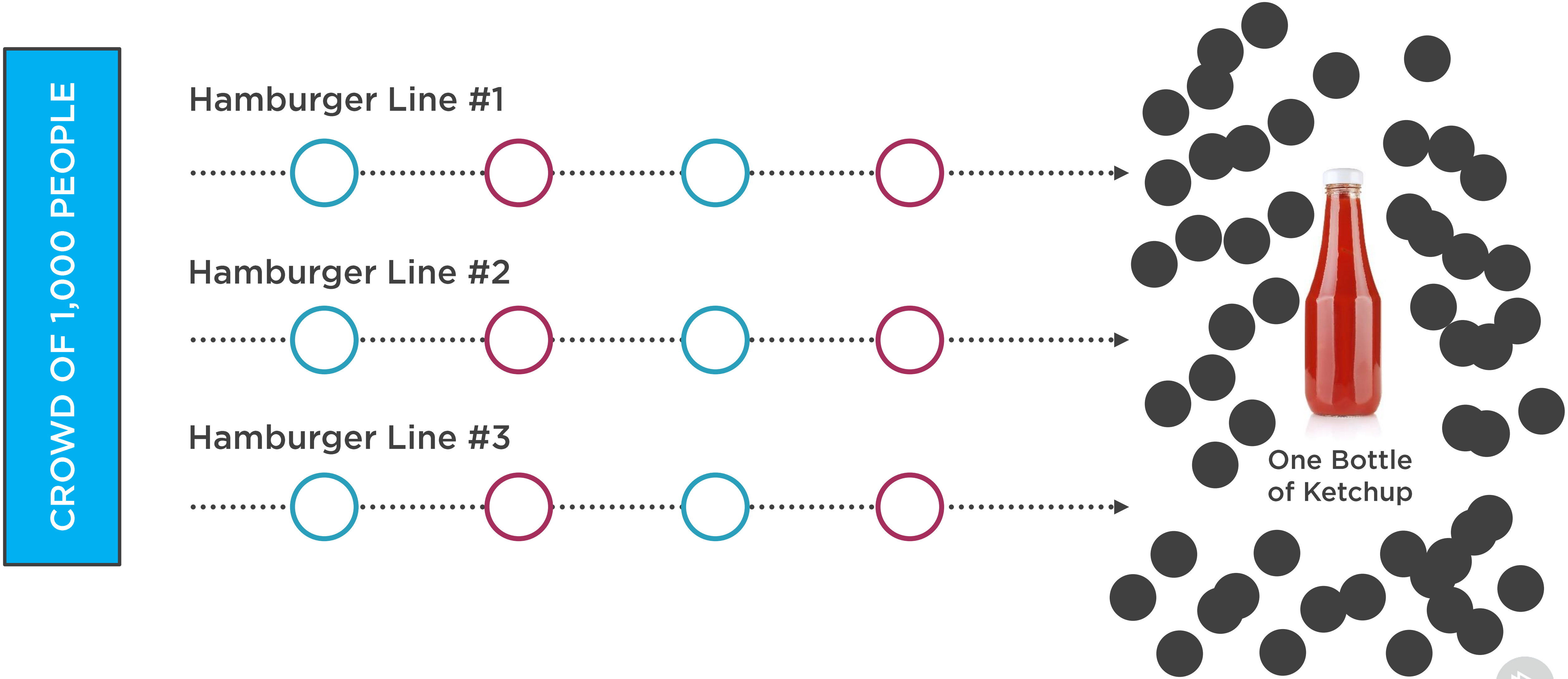
Constraints in the Lunch Line



Constraints in the Lunch Line



Constraints in the Lunch Line



If You Have a Constraint…

…maximize its efficiency

…make sure everything that gets passed to them works on the first try

…make sure they don’t get any rework

The only thing that matters is the constraint.

Summary

DevOps mindset

What is work?

DevOps metrics

Theory of Constraints

Next up:
Planning & Tracking your Team's Work with TFS