



Operations: The Last Mile Problem For DevOps

Damon Edwards
@damonedwards



Damon Edwards



Ops Tools →



Ops Improvement →



Community →

ticketmaster®



EQUIFAX



intuit.



LOYAL3



Cisco
webex



ING

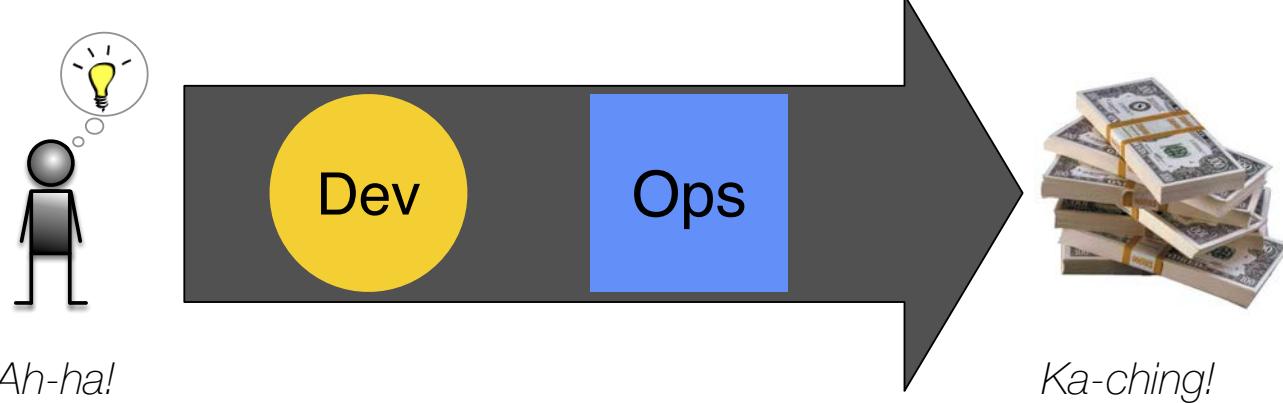
JPMORGAN CHASE & Co.

ETRADE
FINANCIAL

Fidelity
INVESTMENTS

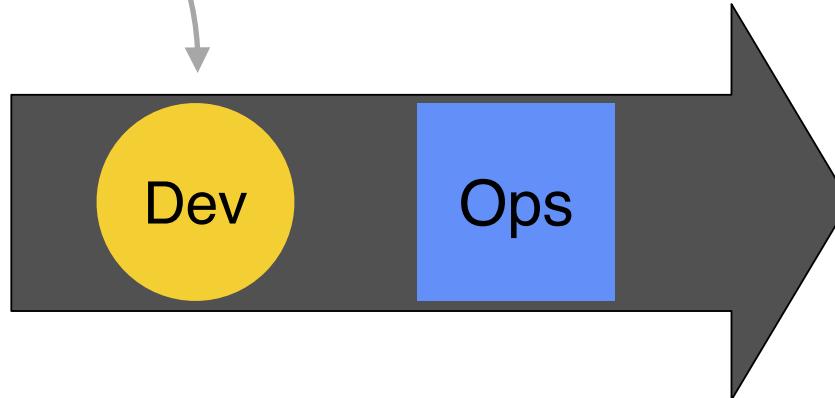
RUN DECK

Developers have had an unfair advantage.





Agile 2011



Ah-ha!



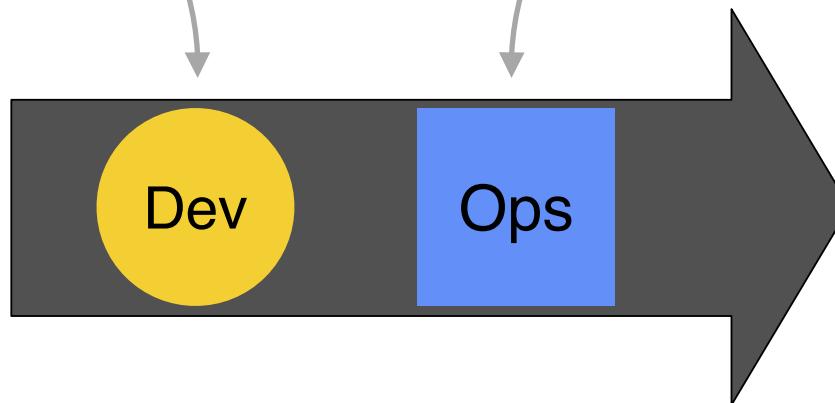
Ka-ching!



Agile
2011



ITIL
1989



Ah-ha!

Ka-ching!

Now here we are in 2018



Let me tell you a true story...



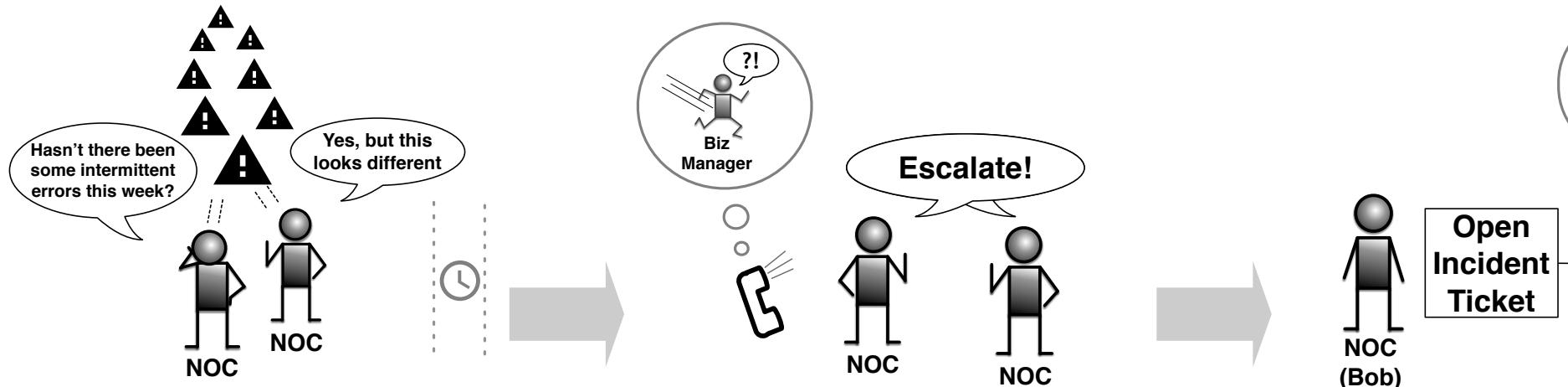
**Digital
Agile
DevOps
SRE**

**Cloud
Docker
Kubernetes
Microservices**



**But nobody was talking about what
happened *after* deployment...**

It was just another Tuesday...



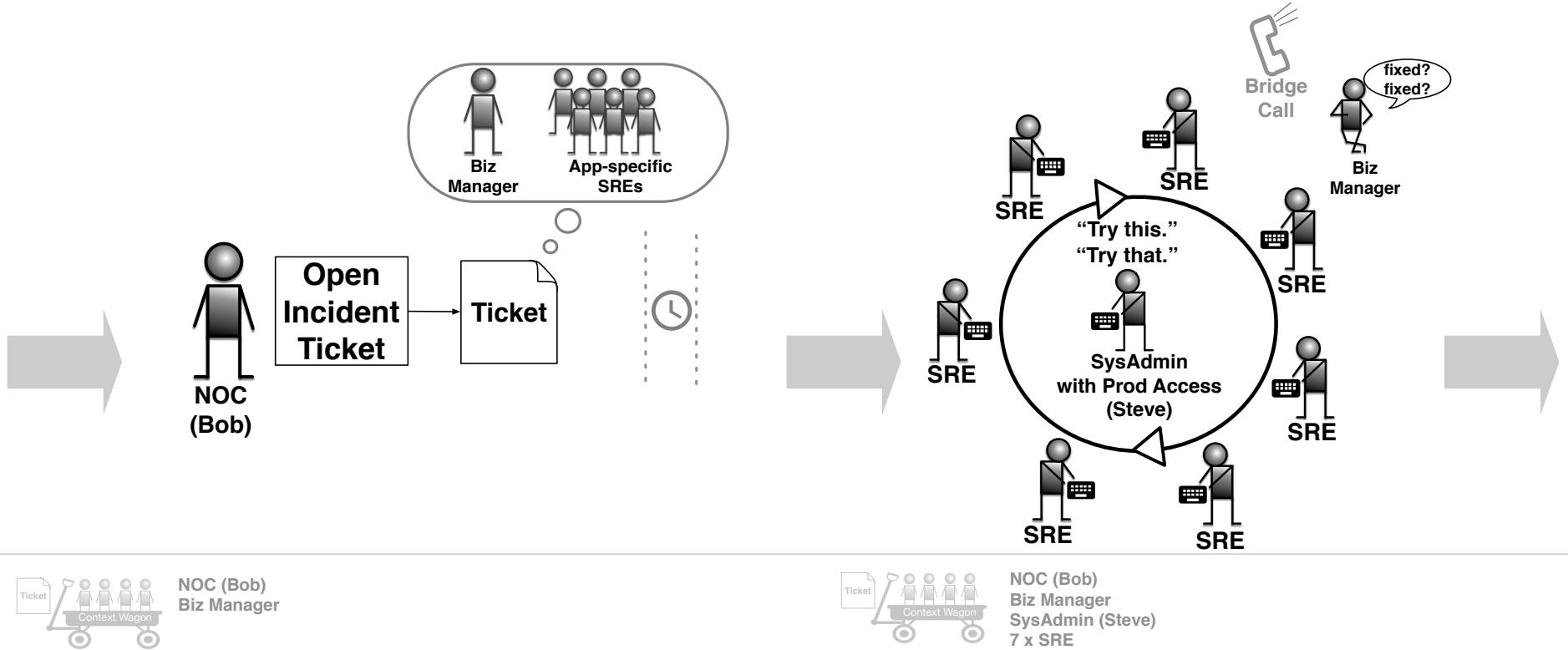
9:30am

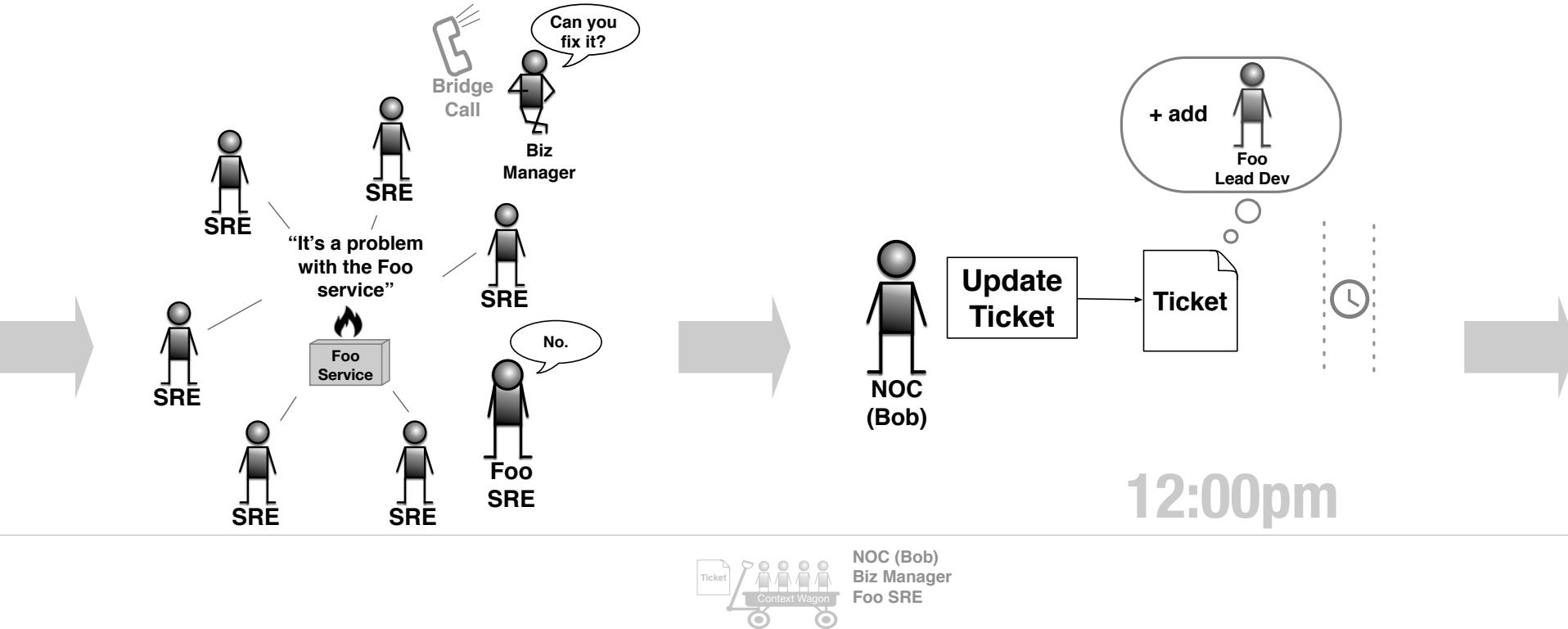
10:00am

v3



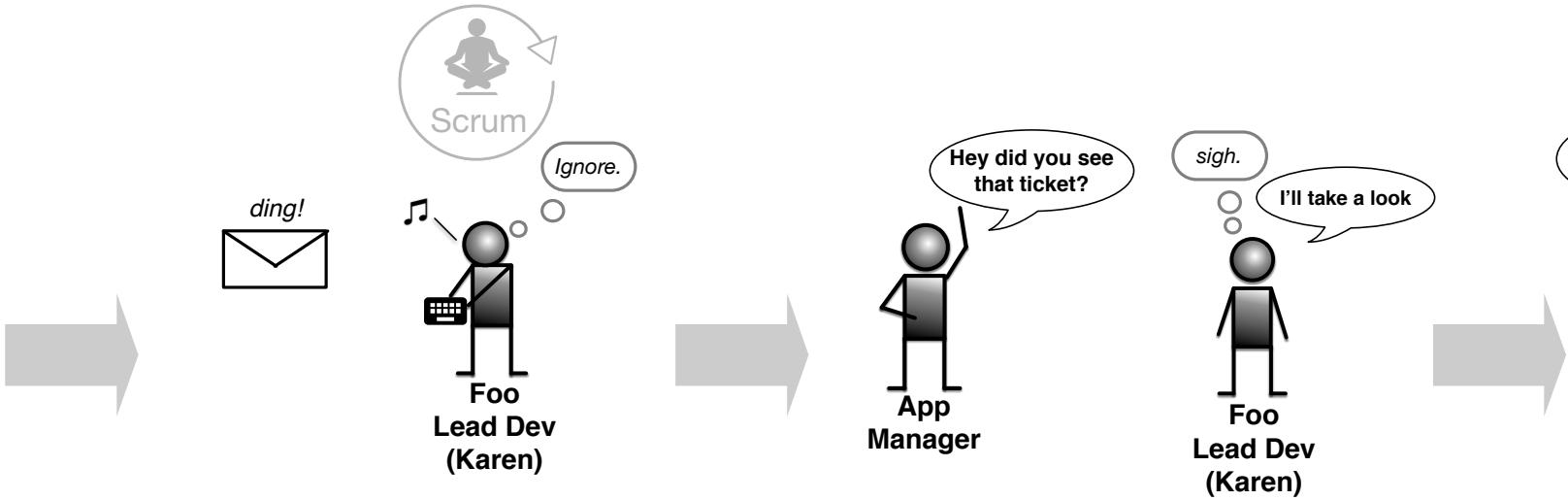
NOC (Bob)
Biz Manager

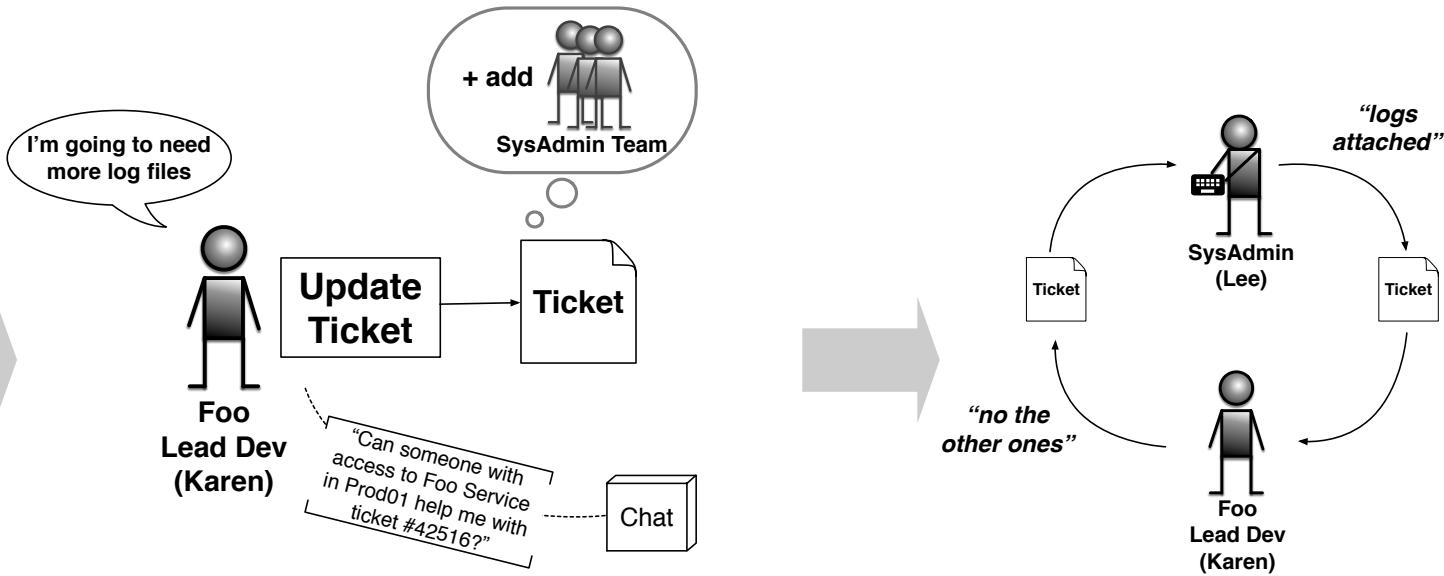






pm

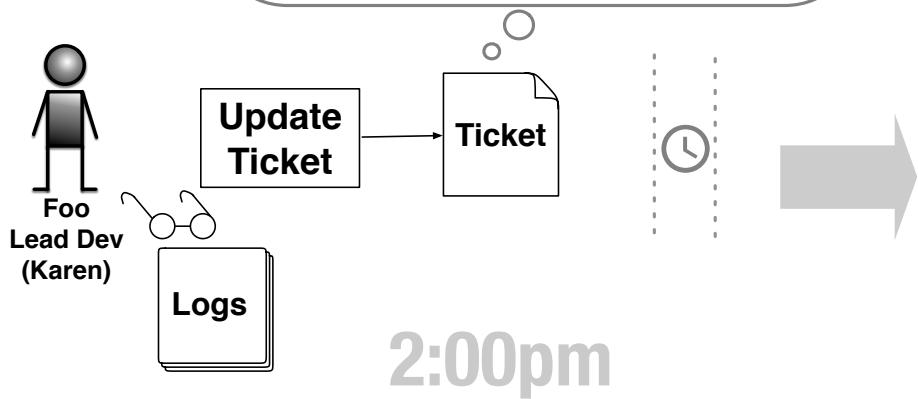




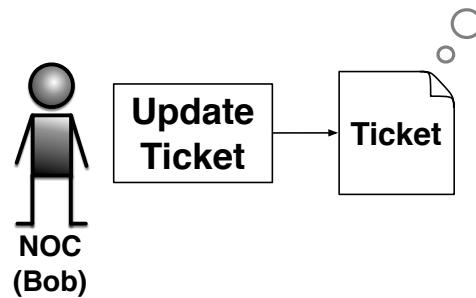
NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)

-Who restarted these services? (and why?)
-They didn't use the correct environment variables!
-This entire service pool needs to be restarted!



+ add Middleware Team
"Middleware, please urgent restart this entire app pool with the correct environment variable"



case
s entire
e correct
able"



NOC
(Bob)



Middleware
Manager
(Melissa)

No way. It's the middle
of the day! You need
business approval.



NOC
(Bob)

Update
Ticket

Ticket

+ add
SVP for Line of
Business



2:30pm

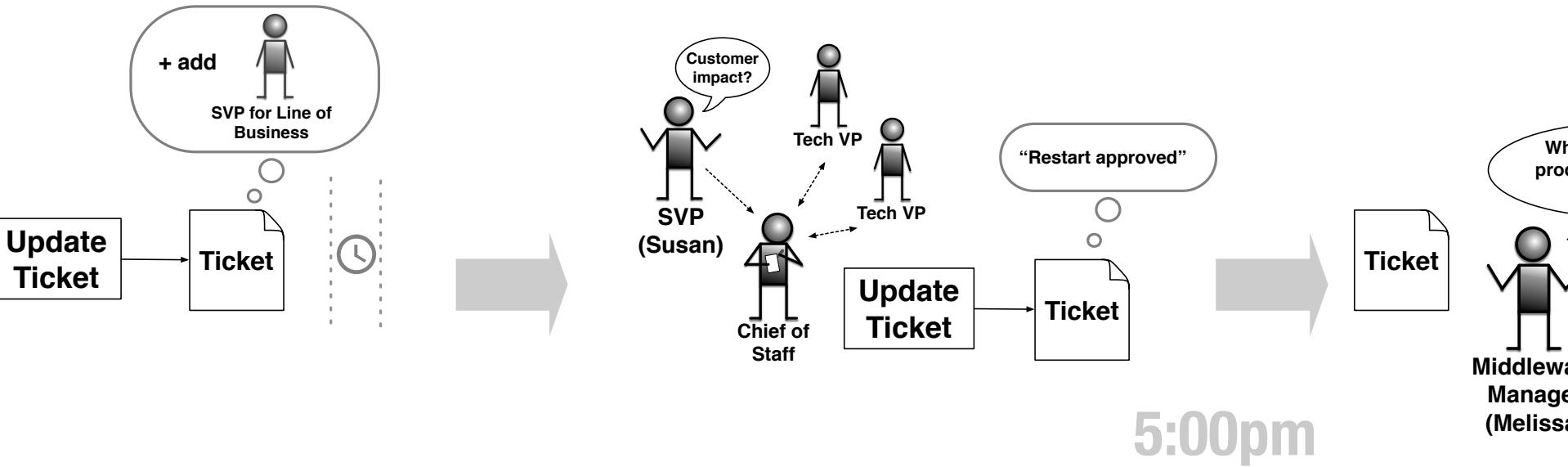


NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager



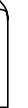
NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE



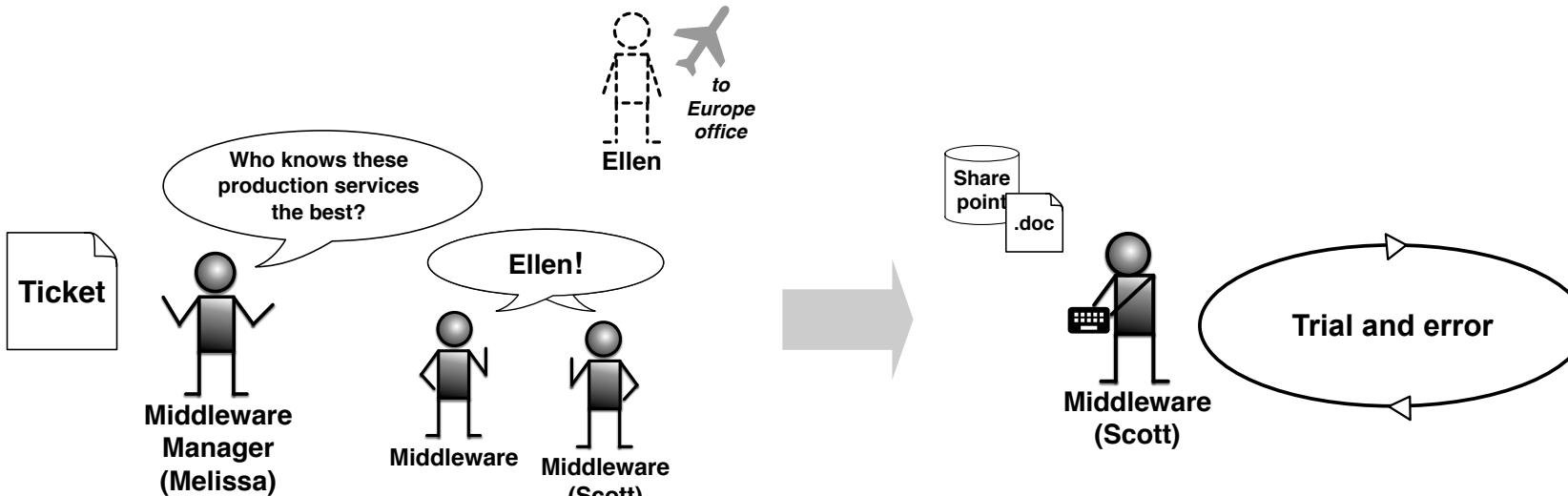
NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

proved"



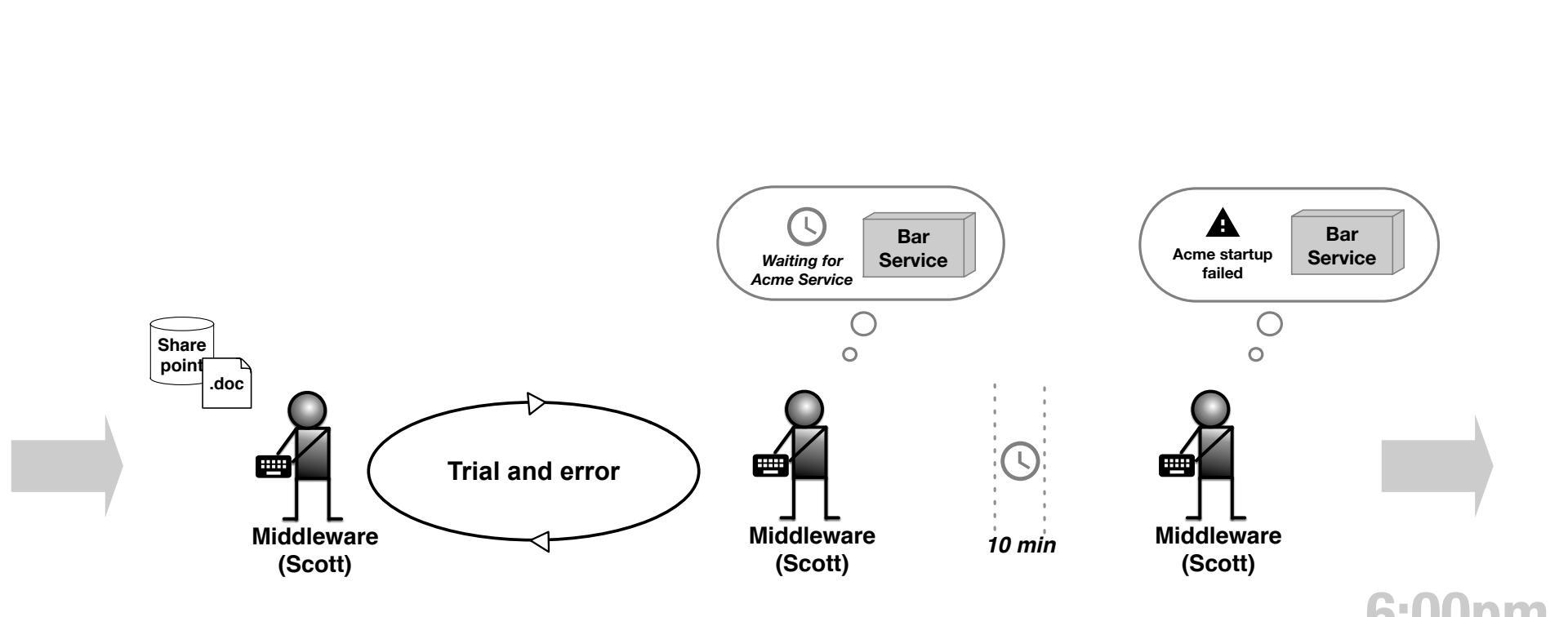
5:00pm



NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

Middleware (Scott)



NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

Middleware (Scott)



Come on.. no.no.no.
What? Why?

Middleware
(Scott)



Come on.. no.no.no.
What? Why?

Middleware (Scott)



Come on.. no.no.no.
What? Why?

Middleware (Scott)

-Bar app startup timed out. Error says can't connect to Acme service.
- I looked at Acme but it seems to be running
-Is this error message correct? Why can't Bar connect?

+ add



The new environment pre-flight check is preventing startup.
Looks like Bar's connection to Acme is being blocked.

+ add



-URGENT: Network connection issue between Bar and Acme



Update Ticket

Ticket



Update Ticket

Ticket

6:45



NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

Middleware (Scott)
Bar SRE (Linda)

The new environment pre-flight check is preventing startup.
Looks like Bar's connection to Acme is being blocked.



-URGENT: Network
connection issue
between Bar and
Acme

+ add



Network
SRE Team

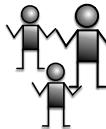
Bar
Lead Dev

I can comment out
the test... But the
CD pipeline only
goes to QA ENV!



Bar
Lead Dev
(Liu)

Customers are
calling. What
is going on?



Business
Managers

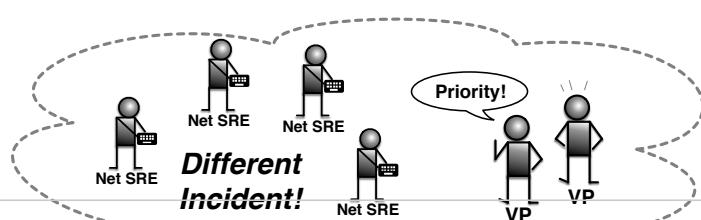
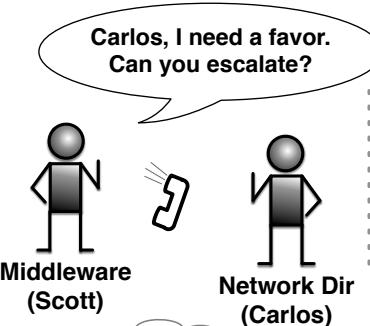
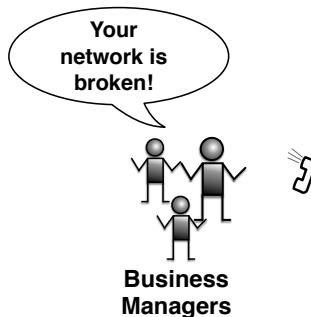
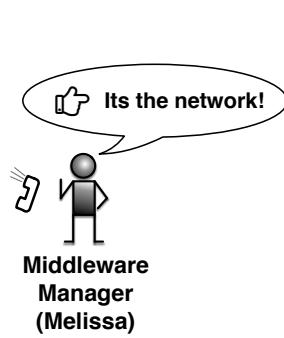


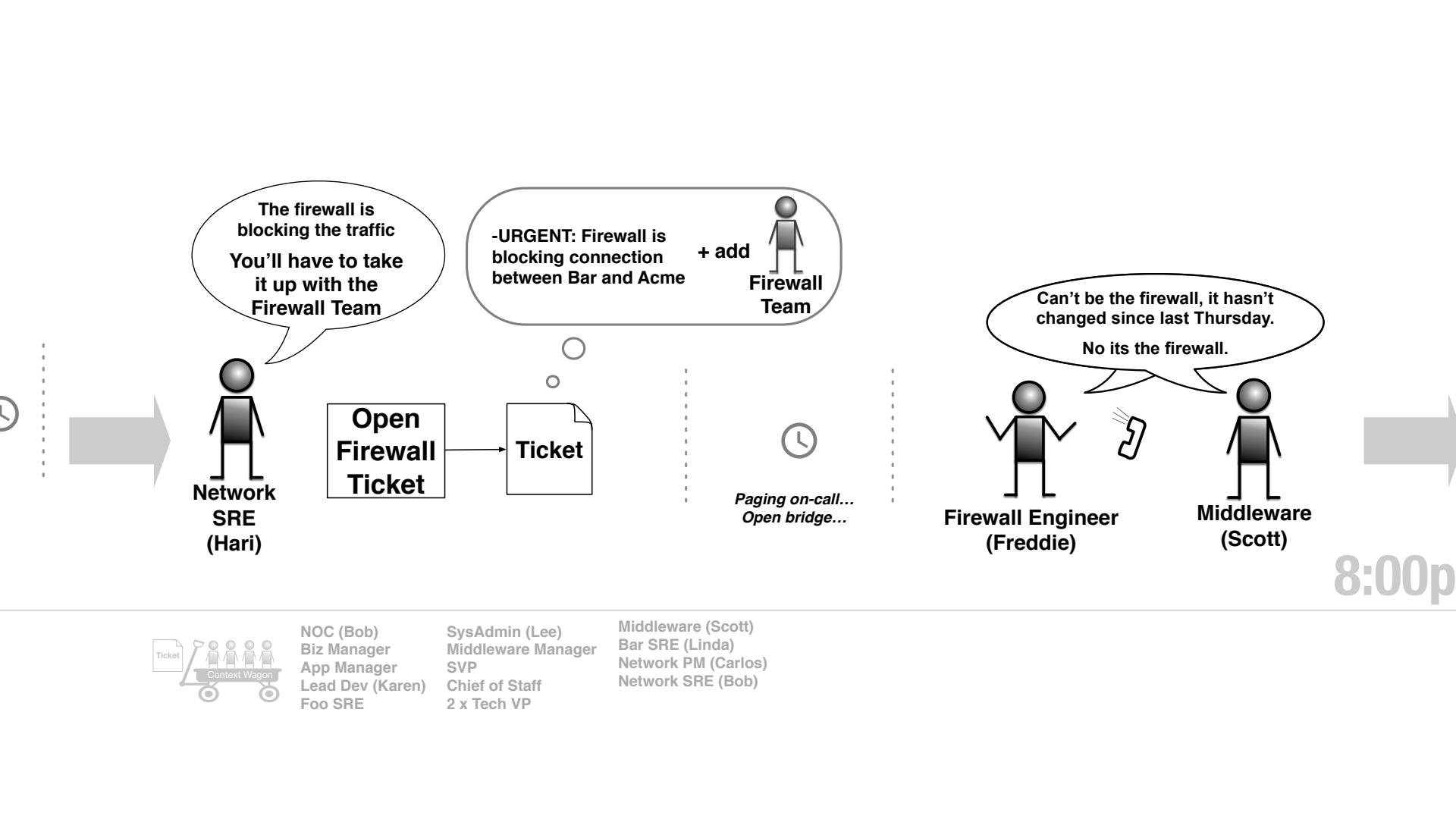
6:45pm

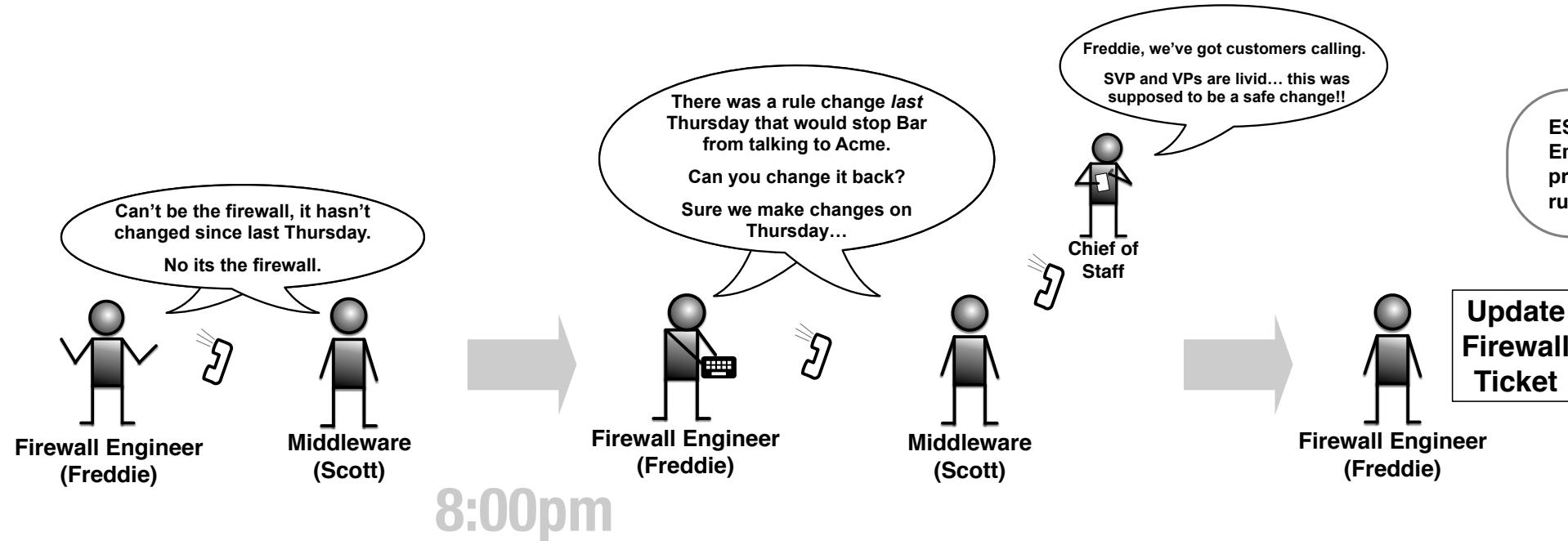
Bob
Manager
Manager
Karen
EVP
Chief of Staff
2 x Tech VP

SysAdmin (Lee)
Middleware Manager
Bar SRE (Linda)
SVP
Chief of Staff
2 x Tech VP

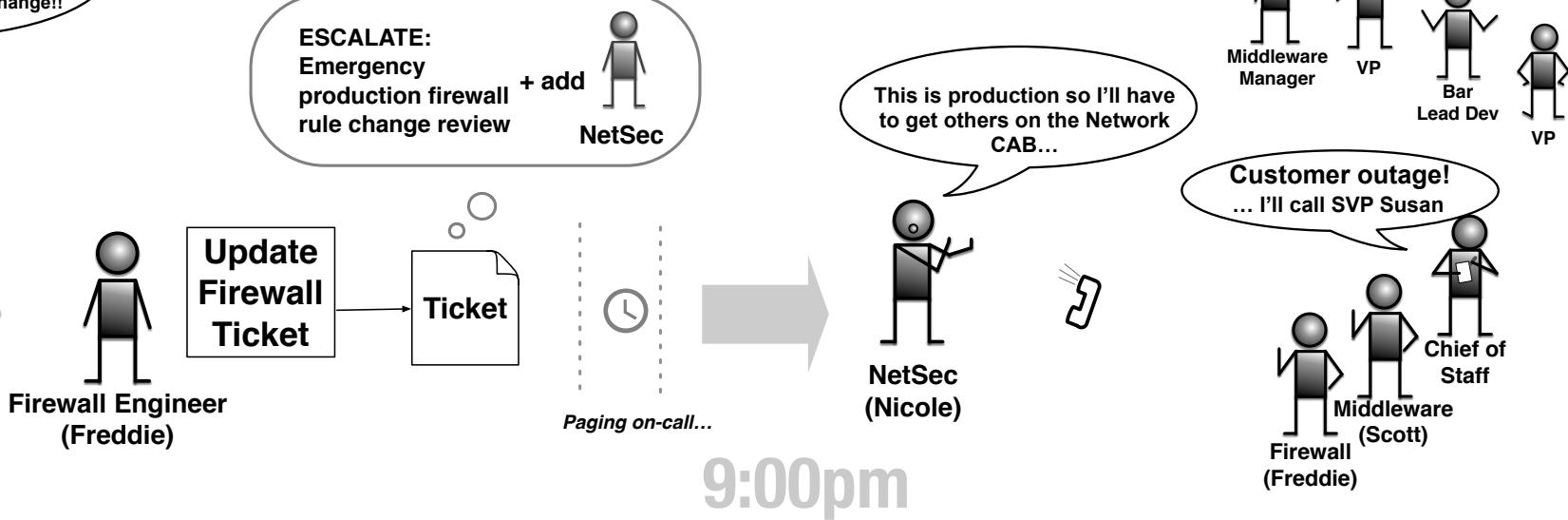
out
the
only
V!





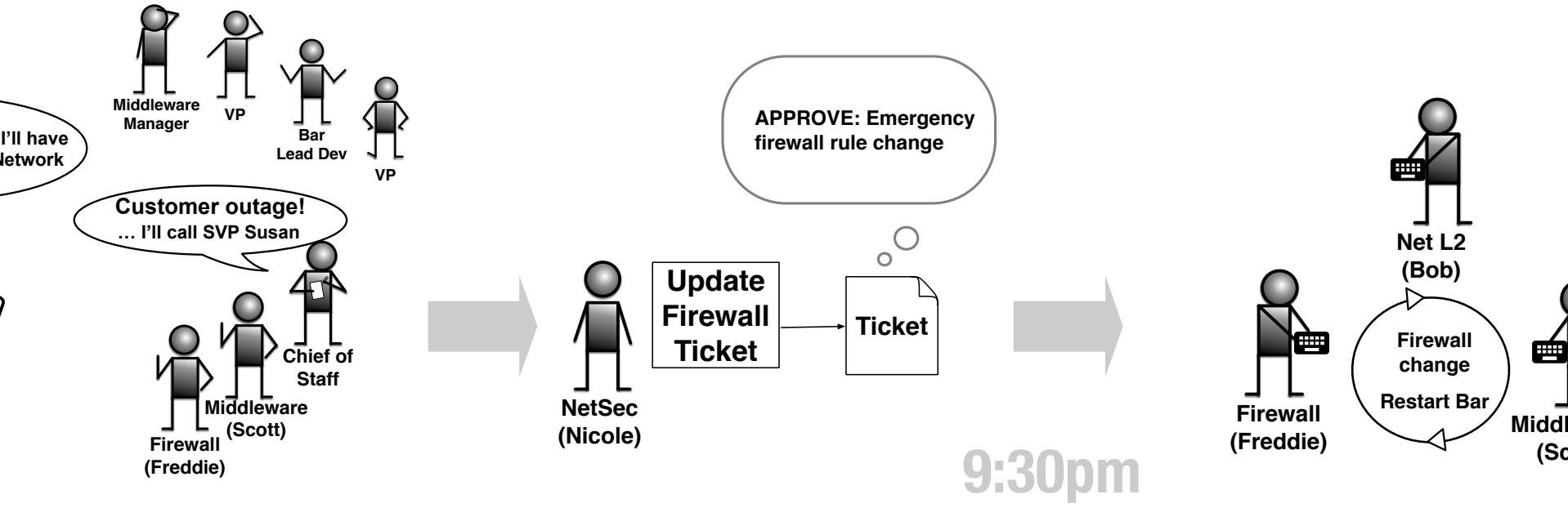


We've got customers calling.
VPs are livid... this was
set to be a safe change!!



NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAd
Middle
SVP
Chief c
2 x Tec

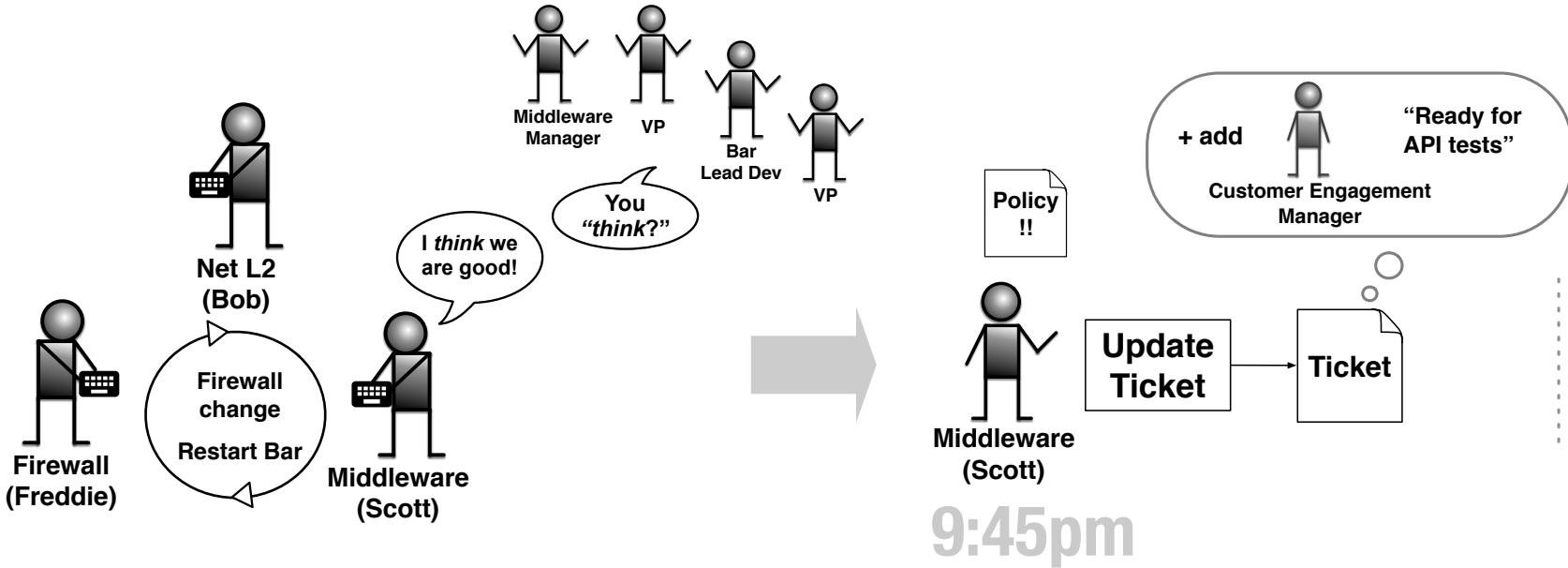


NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

Middleware (Scott)
Bar SRE (Linda)
Network PM (Carlos)
Network SRE (Bob)
Firewall (Freddie)

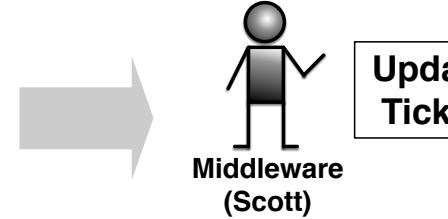
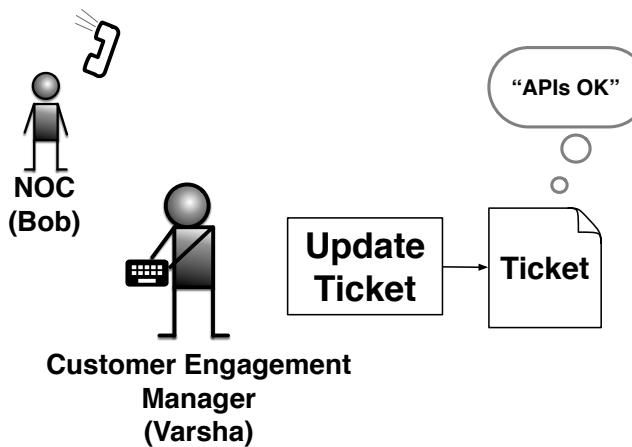
NetSec (Nicole)



“Ready for
API tests”

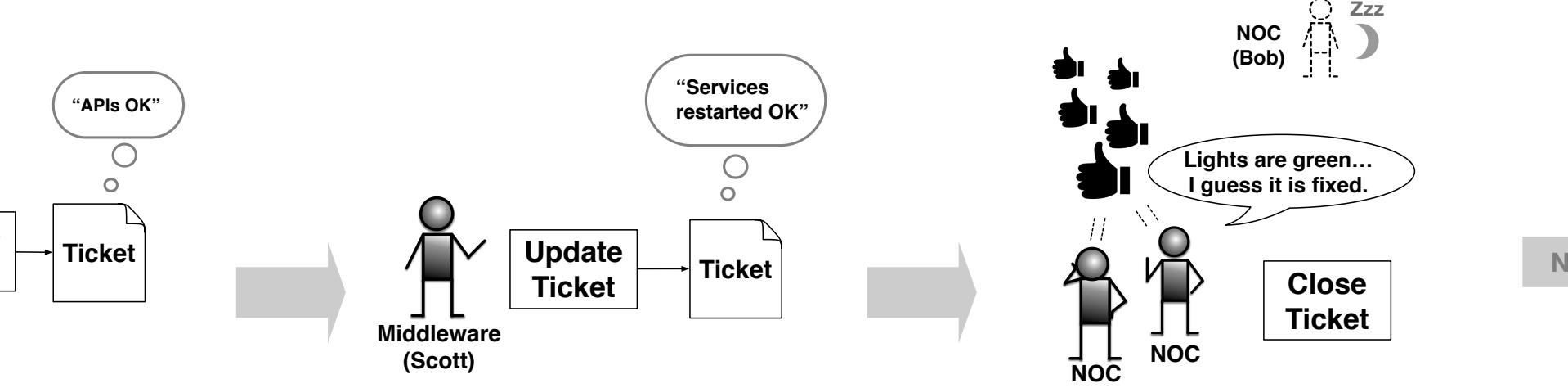
agement

et



11:00pm





11:30pm

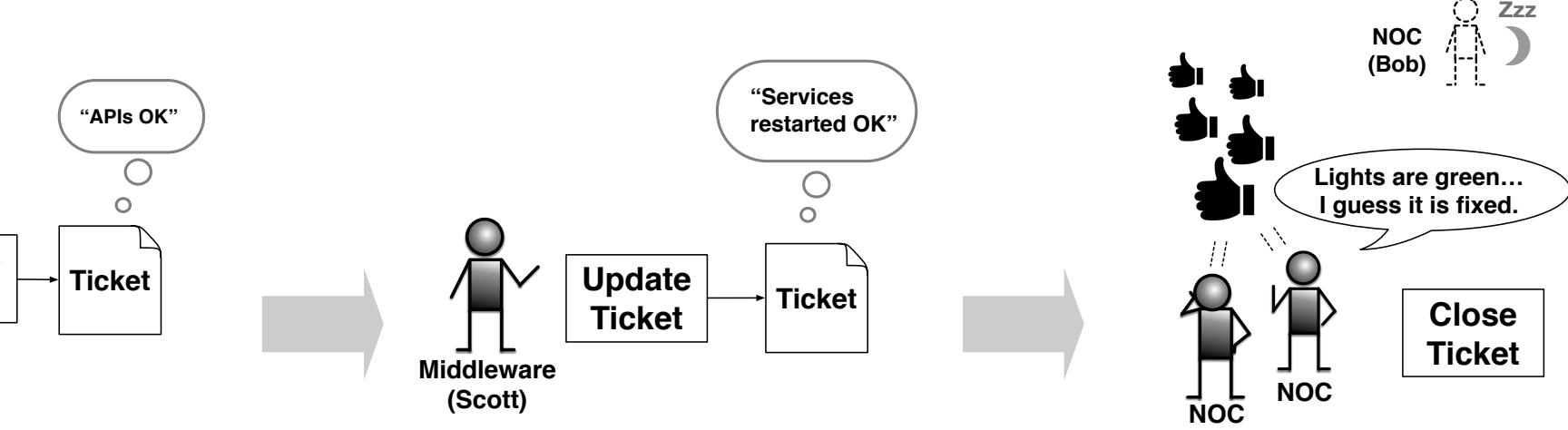


NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

Middleware (Scott)
Bar SRE (Linda)
Network PM (Carlos)
Network SRE (Bob)
Firewall (Freddie)

NetSec (Nicole)
Cust. Engmt. (Varsha)



11:30pm

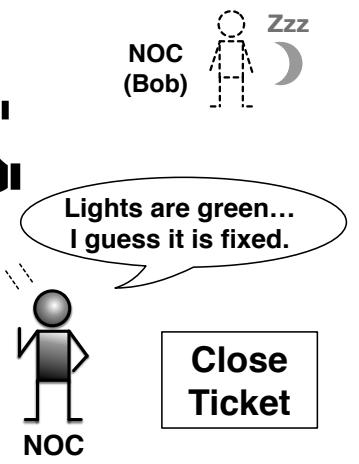


NOC (Bob)
Biz Manager
App Manager
Lead Dev (Karen)
Foo SRE

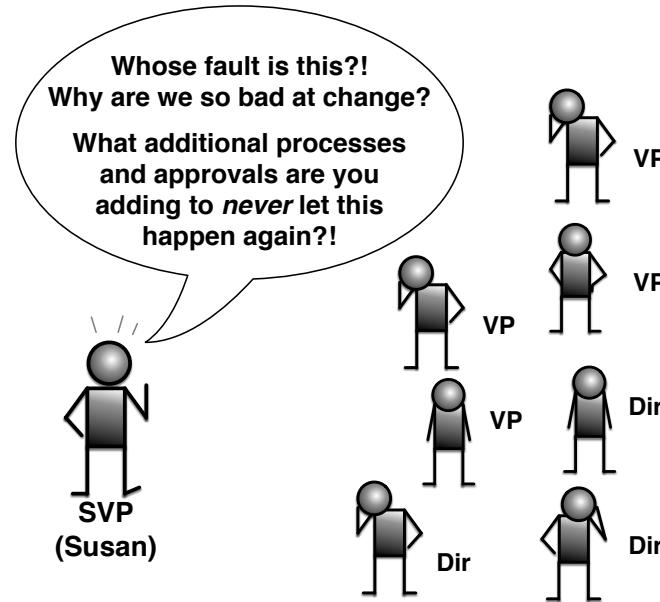
SysAdmin (Lee)
Middleware Manager
SVP
Chief of Staff
2 x Tech VP

Middleware (Scott)
Bar SRE (Linda)
Network PM (Carlos)
Network SRE (Bob)
Firewall (Freddie)

NetSec (Nicole)
Cust. Engmt. (Varsha)



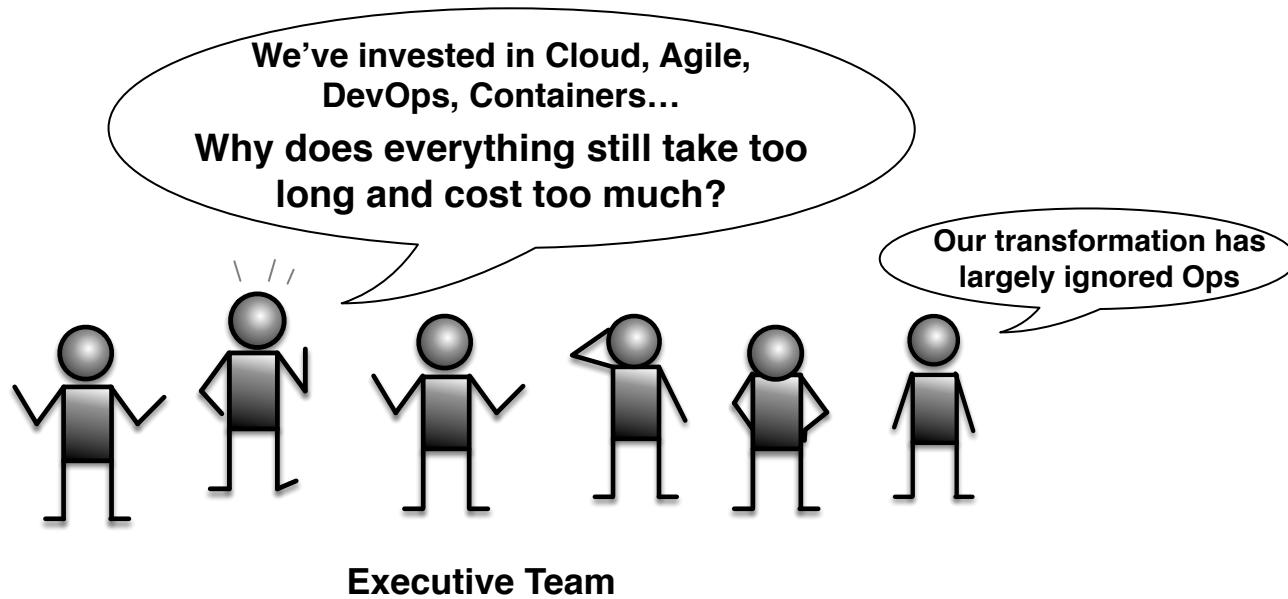
Next Day



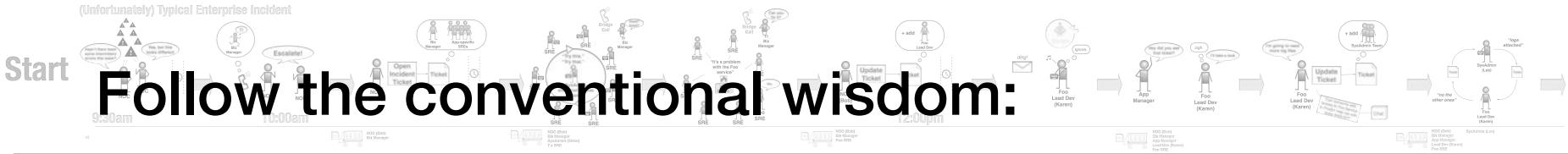
cott)
a)
Carlos)
(Bob)
(die)

NetSec (Nicole)
Cust. Engmt. (Varsha)

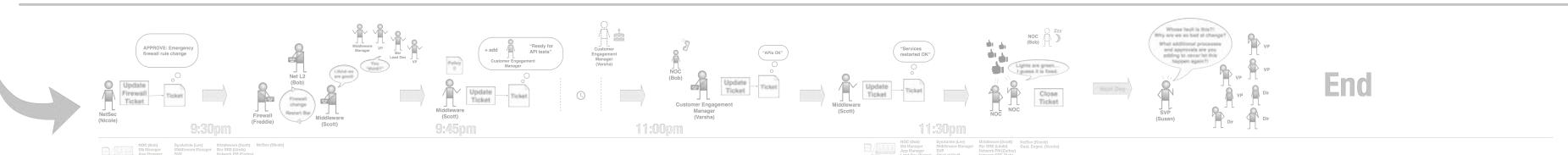
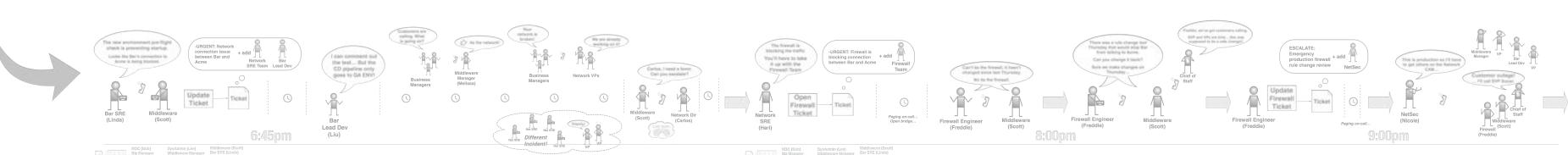
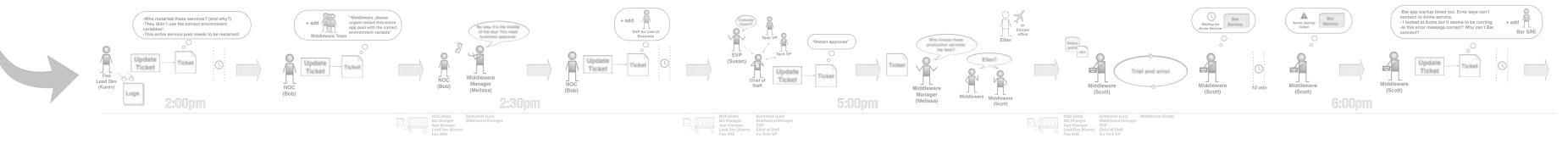
Later...



Most companies chase the symptoms...



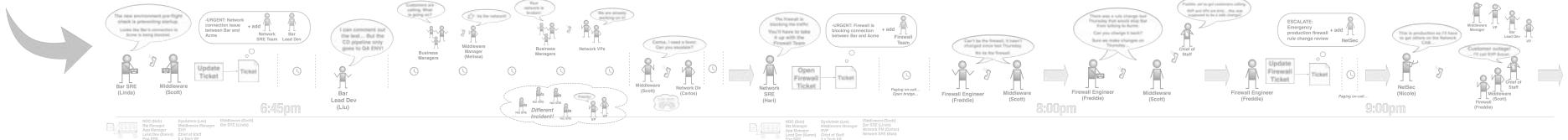
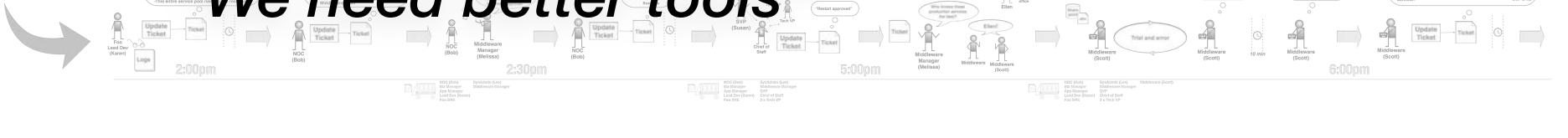
Follow the conventional wisdom:



End



“We need better tools”



(Unfortunately) Typical Enterprise Incident

Start

Follow the conventional wisdom:

"We need better tools"

"We need more people"

End

(Unfortunately) Typical Enterprise Incident

Start

Follow the conventional wisdom:

"We need better tools"

"We need more people"

"We need more discipline and attention to detail"

End

(Unfortunately) Typical Enterprise Incident

Start

Follow the conventional wisdom:

"We need better tools"

"We need more people"

"We need more discipline and attention to detail"

"We need more change reviews/approvals"

End

(Unfortunately) Typical Enterprise Incident

Start

Follow the conventional wisdom:

"We need better tools"

"We need more people"

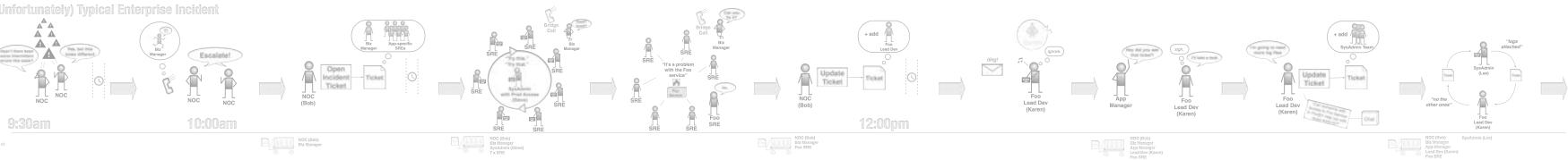
"We need more discipline and attention to detail"

"We need more change reviews/approvals"

End

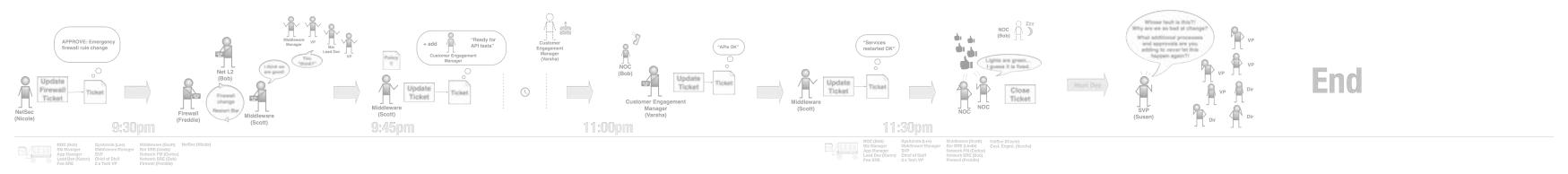
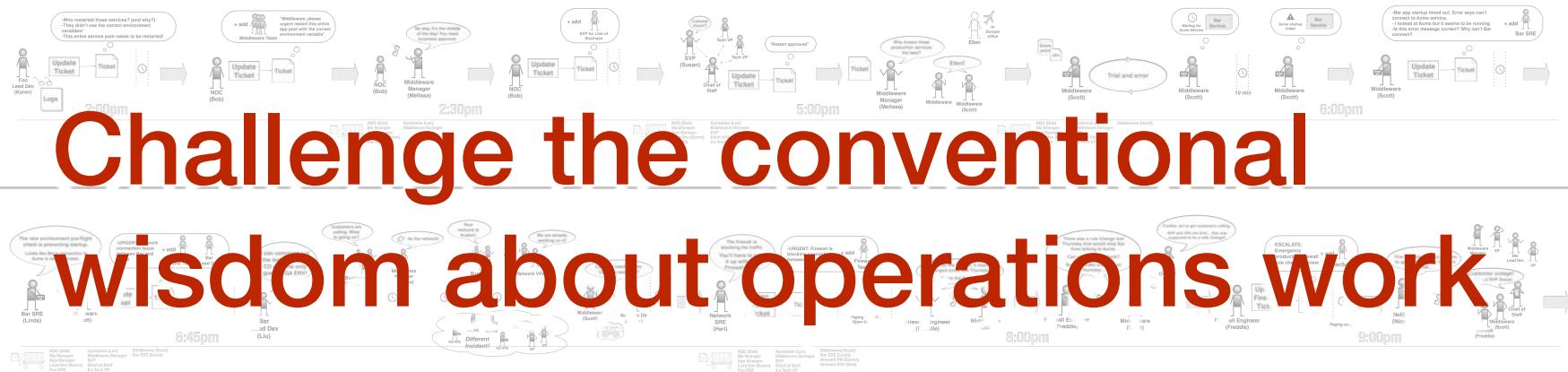
(Unfortunately) Typical Enterprise Incident

Start

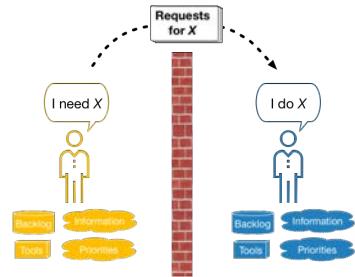


Challenge the conventional wisdom about operations work

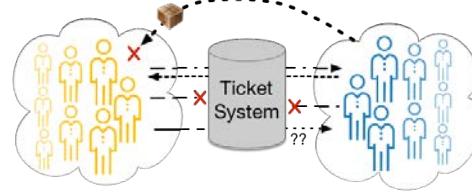
End



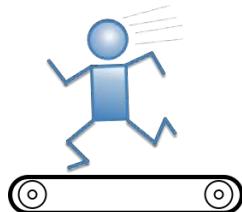
Forces That Undermine Operations



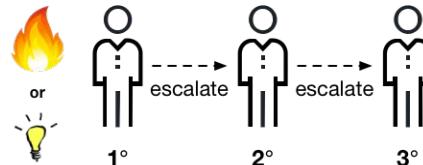
Silos



Queues

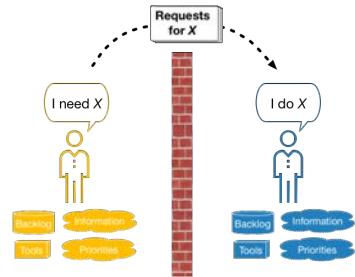


Toil

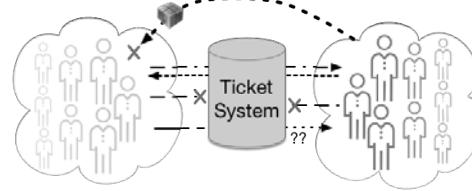


Low Trust

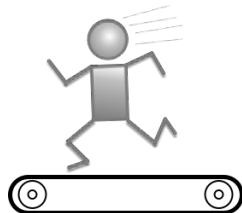
Forces That Undermine Operations



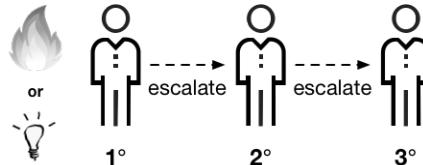
Silos



Queues

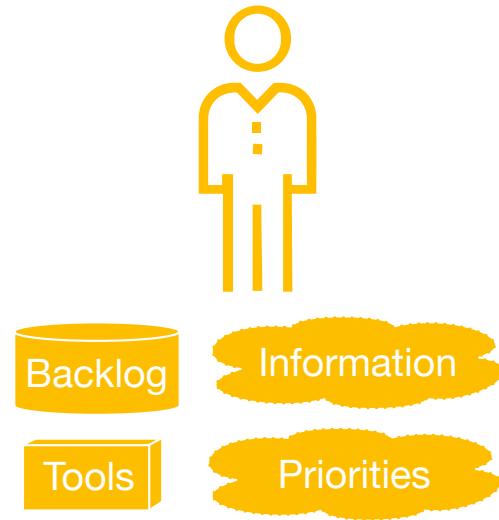


Toil



Low Trust

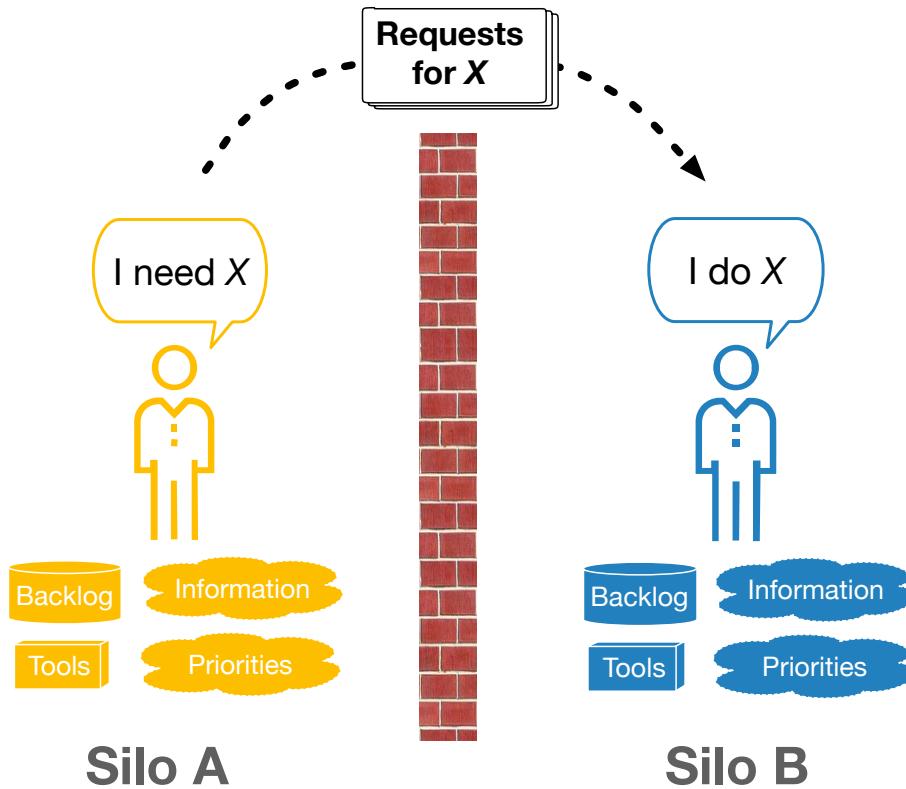
Silos



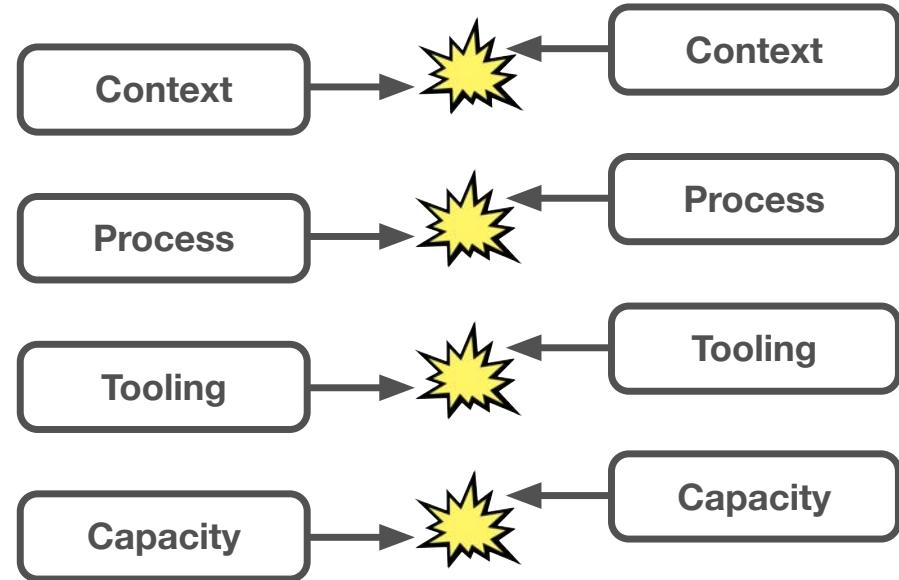
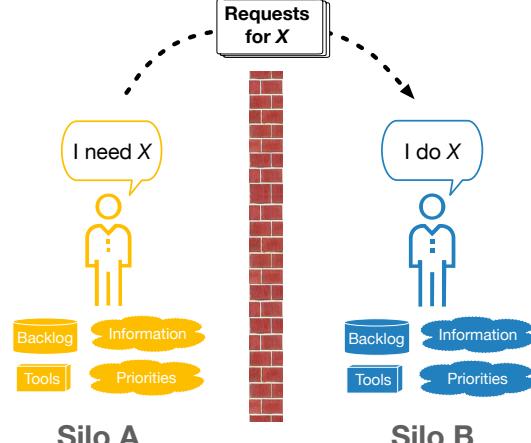
Silos



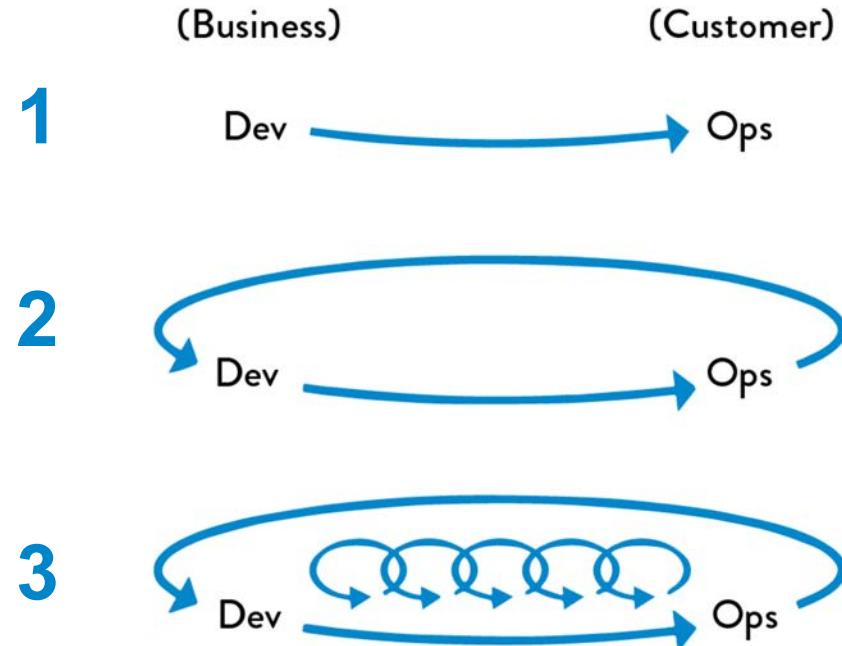
Silos



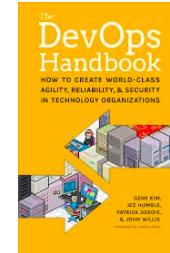
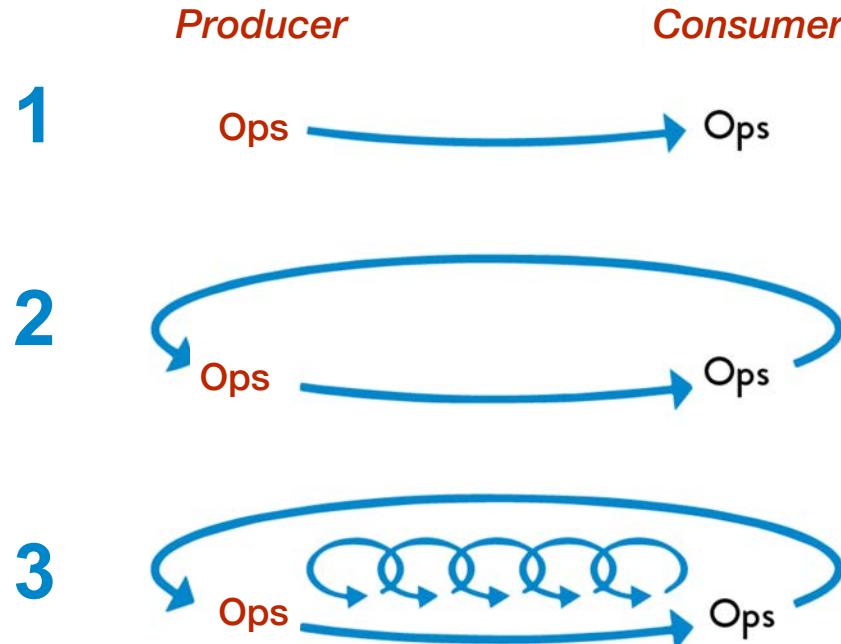
Silos cause disconnects and mismatches



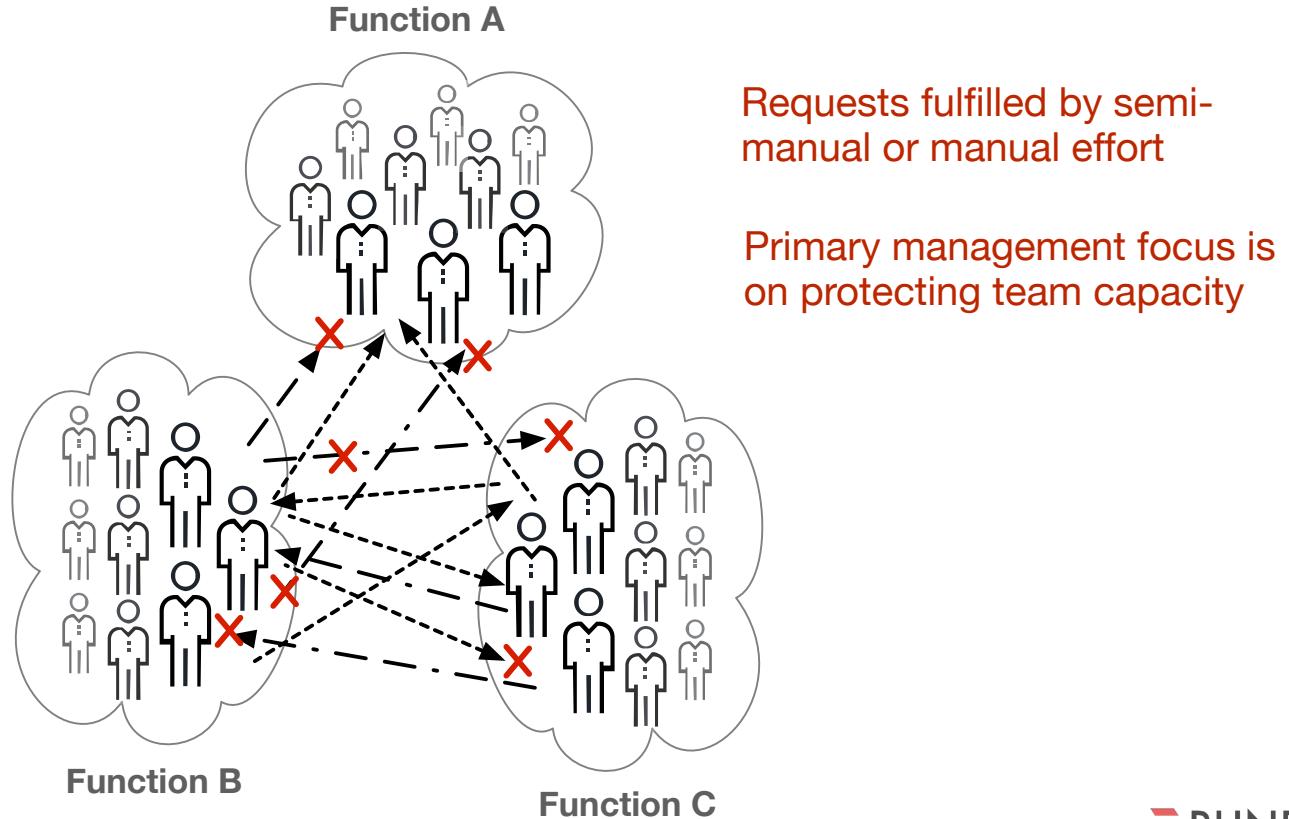
Silos Interfere with feedback loops



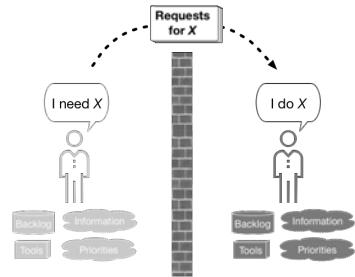
Silos Interfere with feedback loops



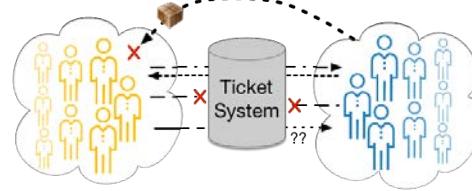
Silos create labor pools of functional specialists



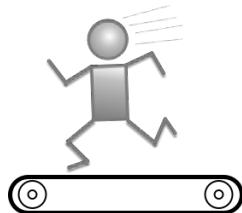
Forces That Undermine Operations



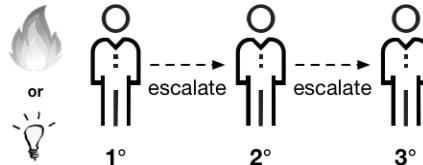
Silos



Queues



Toil

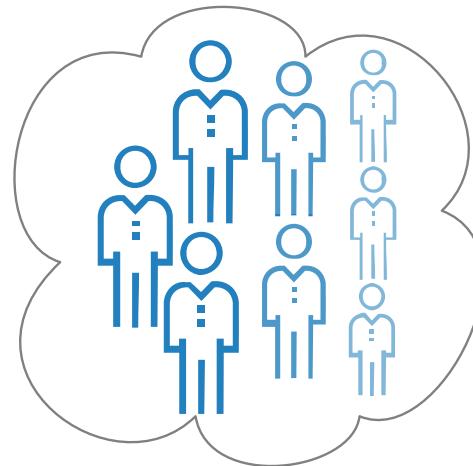


Low Trust

How do we cover for our silos disconnects and mismatches?



Silo A

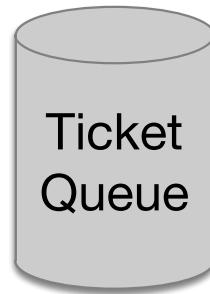


Silo B

How do we cover for our silos disconnects and mismatches?

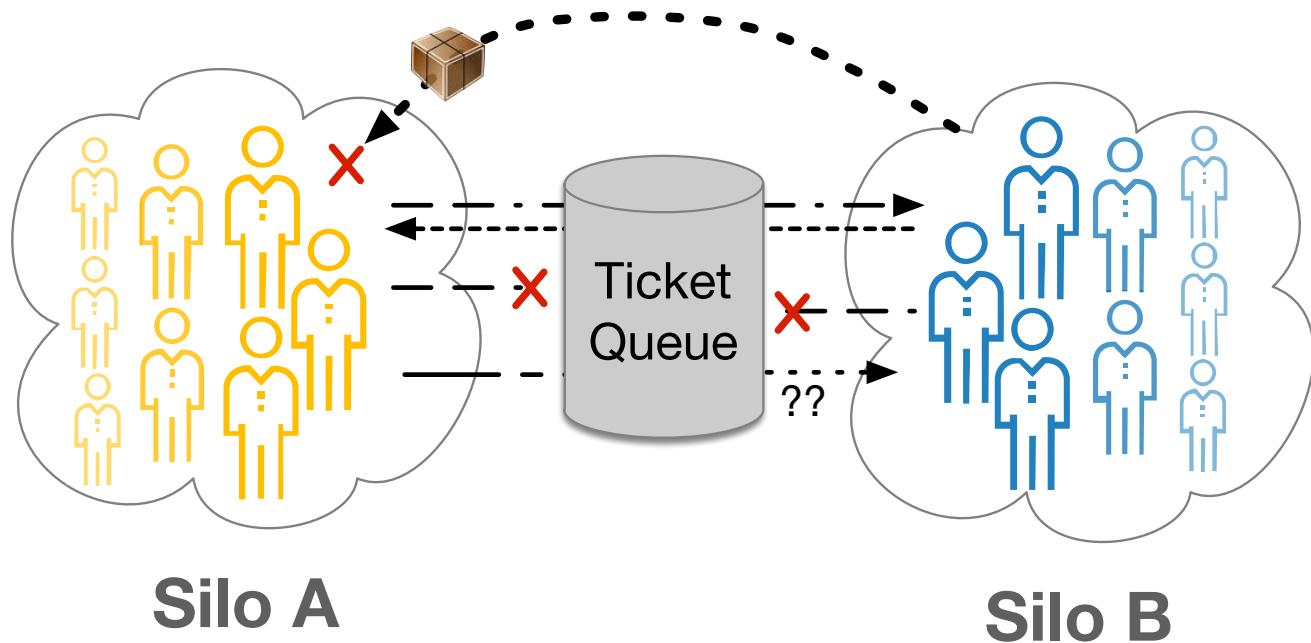


Silo A

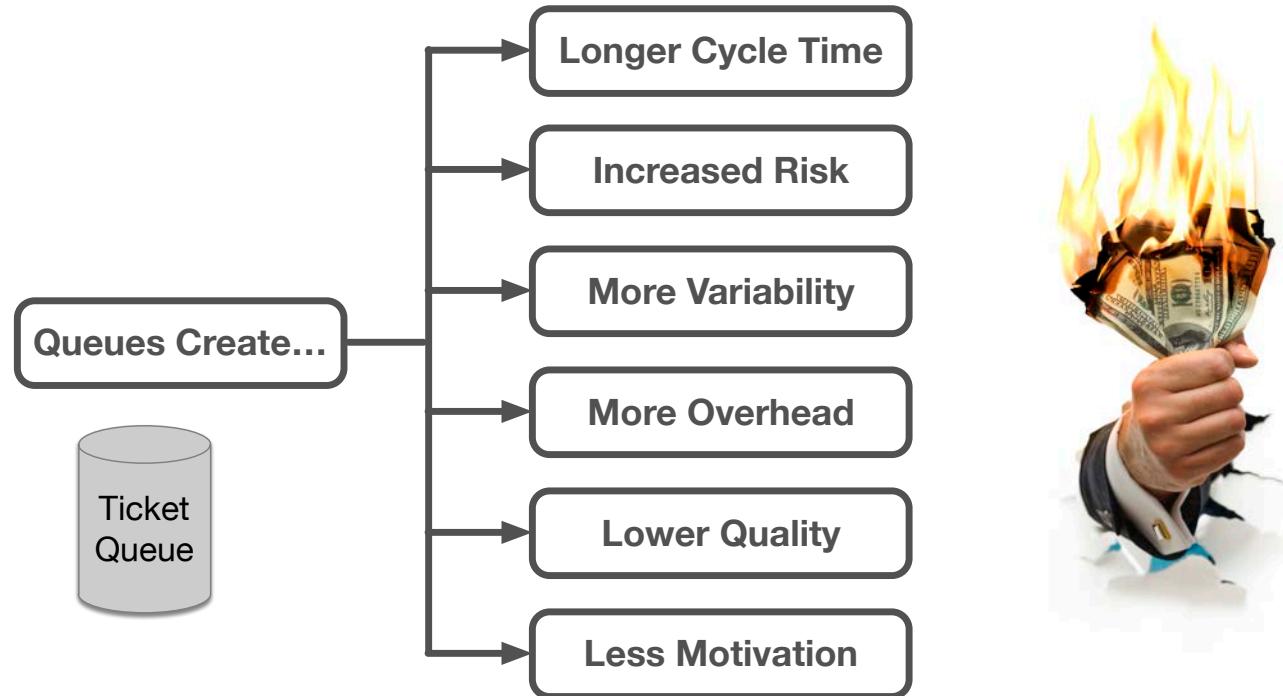


Silo B

We all know how well that works



Request queues are an expensive way to manage work

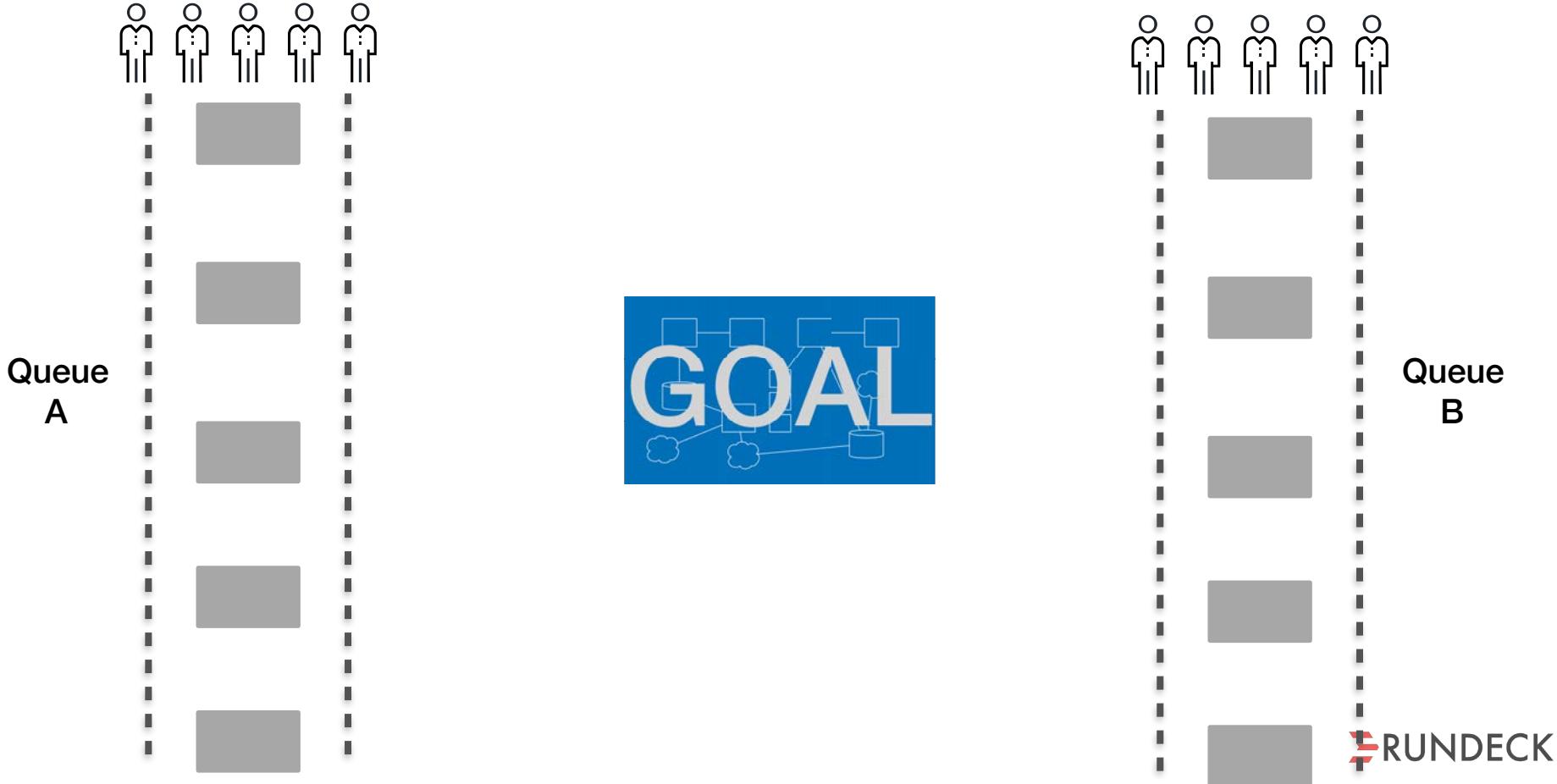


Adapted from Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development

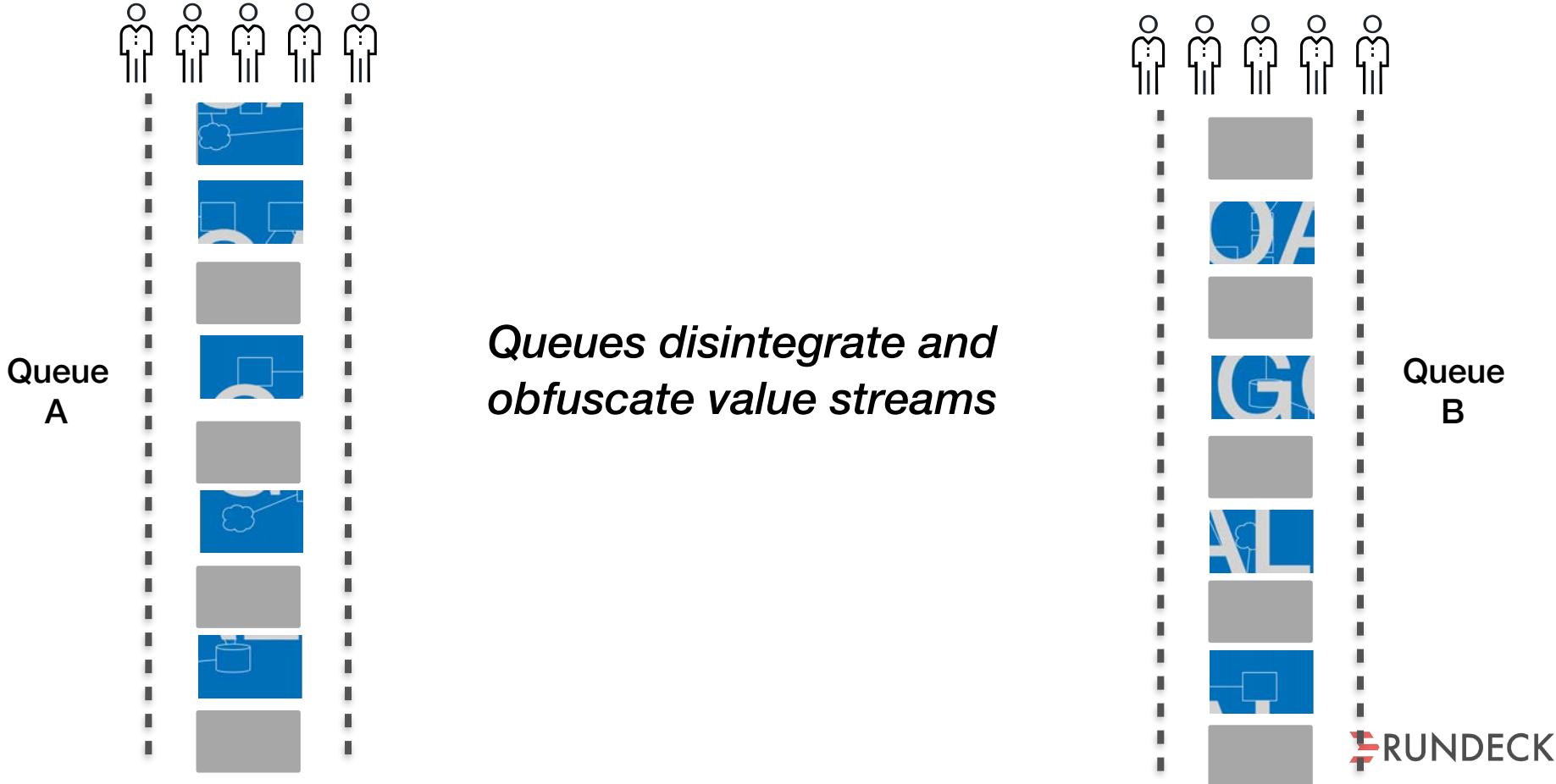
What do queues do to value streams?



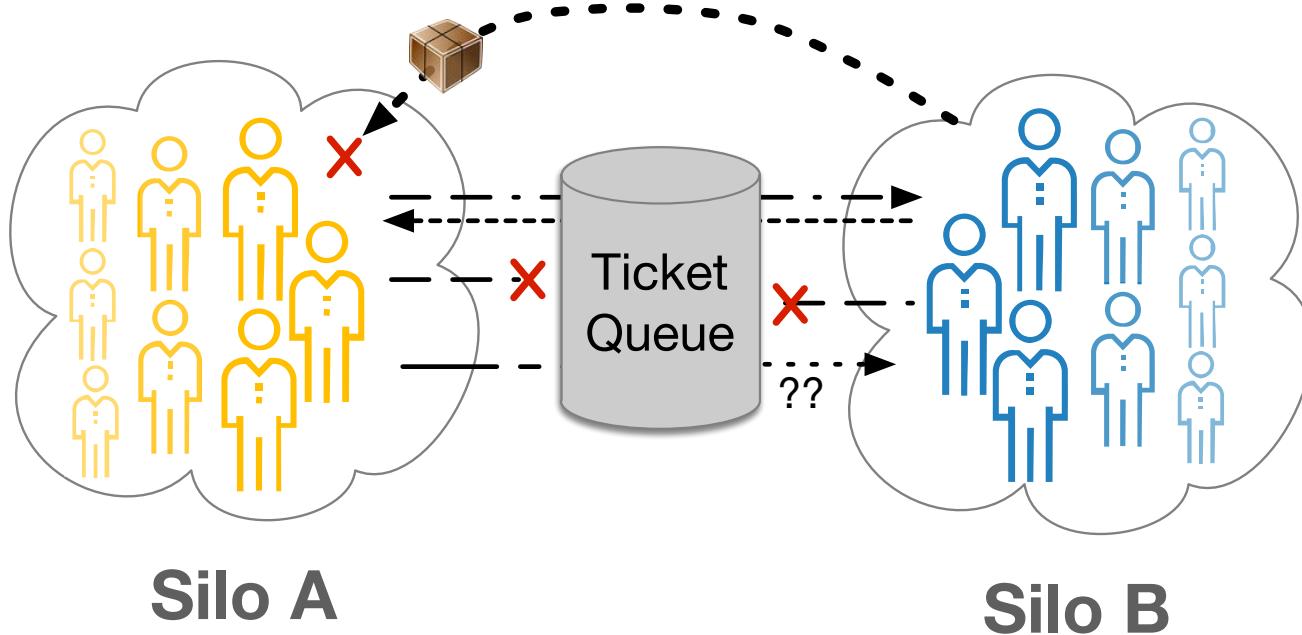
What do queues do to value streams?



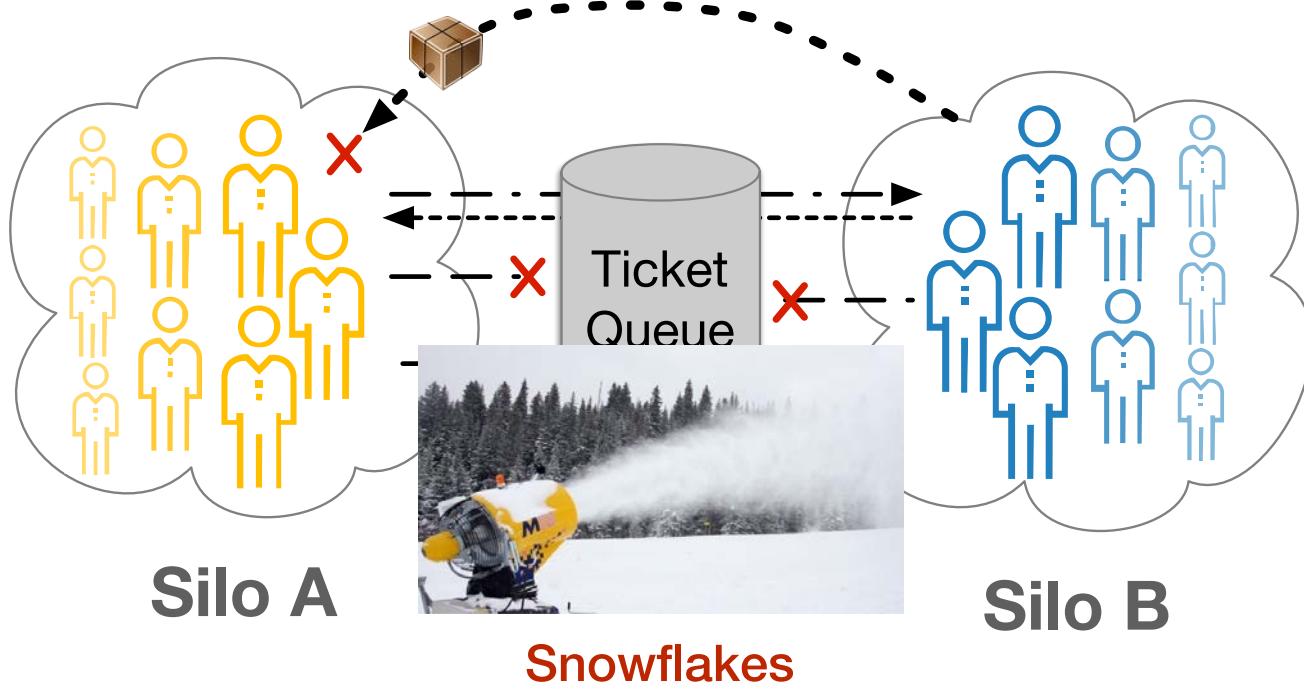
What do queues do to value streams?



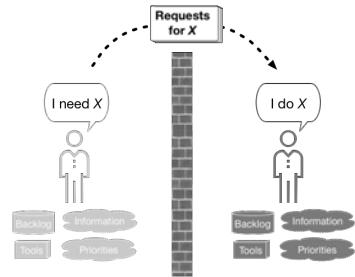
Tickets queues become “snowflake makers”



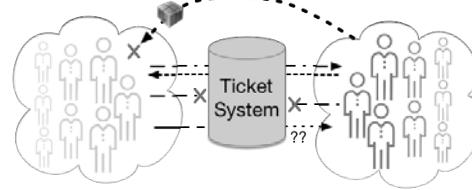
Tickets queues become “snowflake makers”



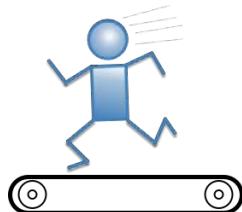
Forces That Undermine Operations



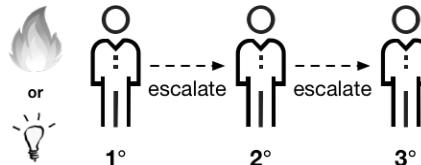
Silos



Queues

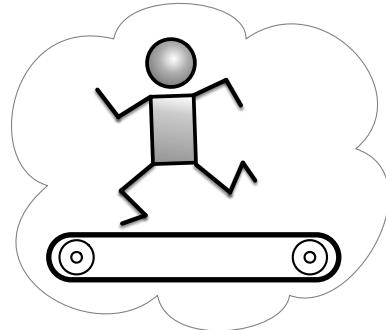


Toil

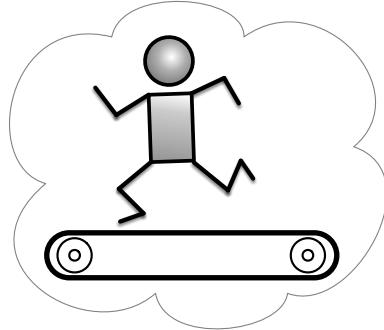


Low Trust

Excessive toil prevents fixing the system



Excessive toil prevents fixing the system

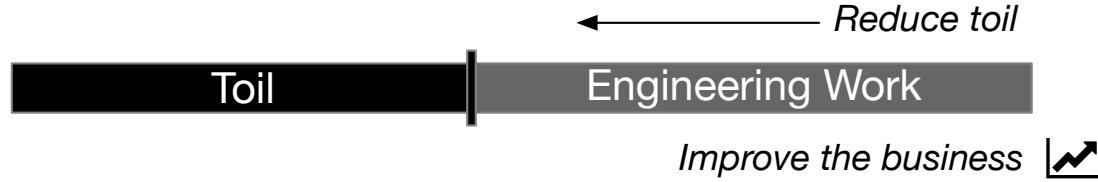


“Toil is the kind of work tied to running a production service that tends to be **manual, repetitive, automatable, tactical, devoid of enduring value, and that scales linearly as a service grows.**”

-Vivek Rau
Google

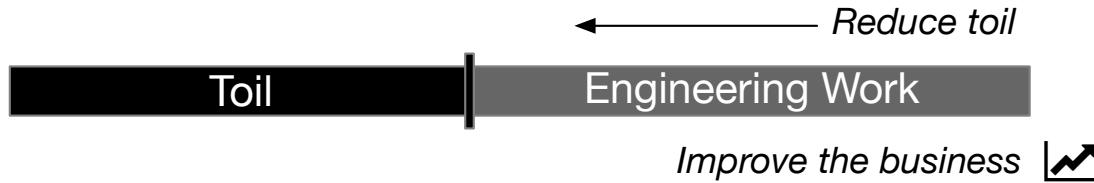
Excessive toil prevents fixing the system

Toil at manageable percentage of capacity

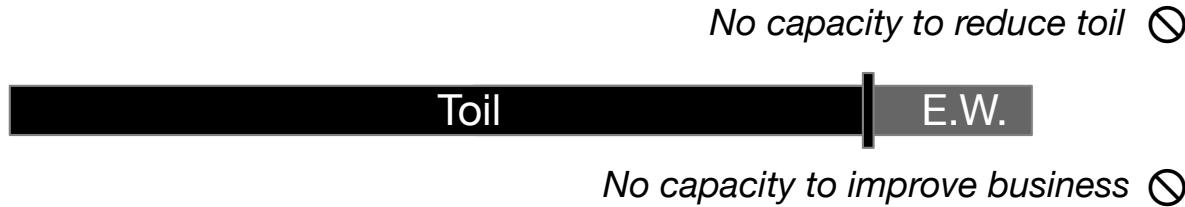


Excessive toil prevents fixing the system

Toil at manageable percentage of capacity



Toil at unmanageable percentage of capacity (“Engineering Bankruptcy”)



Toil Impacts Development As Well

Toil Impacts Development As Well

- 2016-2017 study of development teams
- 14 enterprises (insurance, healthcare, finance, travel, retail)
- Tech org headcount: 900 - 6800



Toil Impacts Development As Well

- 2016-2017 study of development teams
- 14 enterprises (insurance, healthcare, finance, travel, retail)
- Tech org headcount: 900 - 6800



“28% - 63% of development teams’ total time was consumed by operations toil.”

Toil Impacts Development As Well

- 2016-2017 study of development teams
- 14 enterprises (insurance, healthcare, finance, travel, retail)
- Tech org headcount: 900 - 6800



“28% - 63% of development teams’ total time was consumed by operations toil.”

Waiting for environments

Rework due to env. differences

Network issues

Broken lower environments

Incident escalations

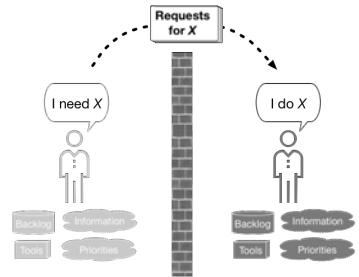
Handoffs

Requests for information

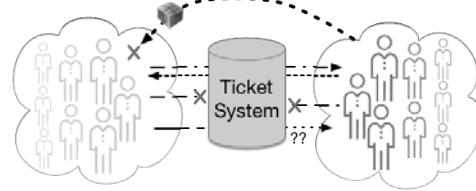
Change meetings

And more...

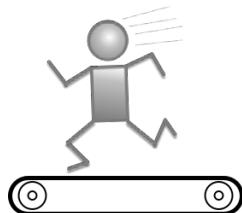
Forces That Undermine Operations



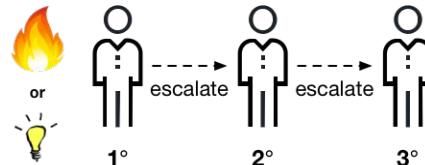
Silos



Queues

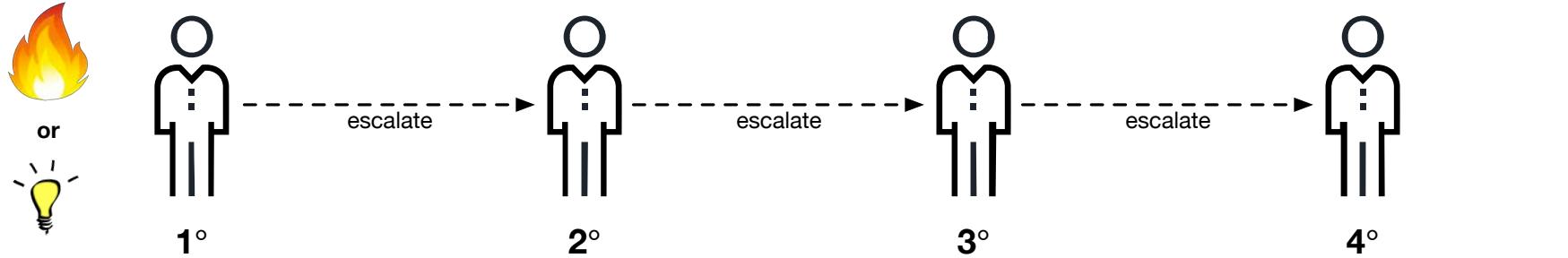


Toil



Low Trust

Where are decisions made? Who can take action?



All work is contextual



John
Allspaw

All work is contextual

```
rm -rf $PATHNAME
```

John
Allspaw



RUNDECK

All work is contextual

```
rm -rf $PATHNAME
```

Is this dangerous?



John
Allspaw

 RUNDECK

All work is contextual

```
rm -rf $PATHNAME
```

John
Allspaw



RUNDECK

All work is contextual

```
rm -rf $PATHNAME
```

```
@@ -1,2 +1,2 @@
-<!-- Status: 0k --> +<!-- Status: OK -->
```

John
Allspaw



RUNDECK

All work is contextual

```
rm -rf $PATHNAME
```

```
@@ -1,2 +1,2 @@
--<!-- Status: 0k --> +<!-- Status: 0K -->
```

Is this dangerous?



John
Allspaw

RUNDECK

All work is contextual

```
rm -rf $PATHNAME
```

```
@@ -1,2 +1,2 @@
-<!-- Status: 0k --> +<!-- Status: OK -->
```

John
Allspaw



RUNDECK

All work is contextual

```
rm -rf $PATHNAME
```

```
@@ -1,2 +1,2 @@
--<!-- Status: 0k --> +<!-- Status: OK -->
```

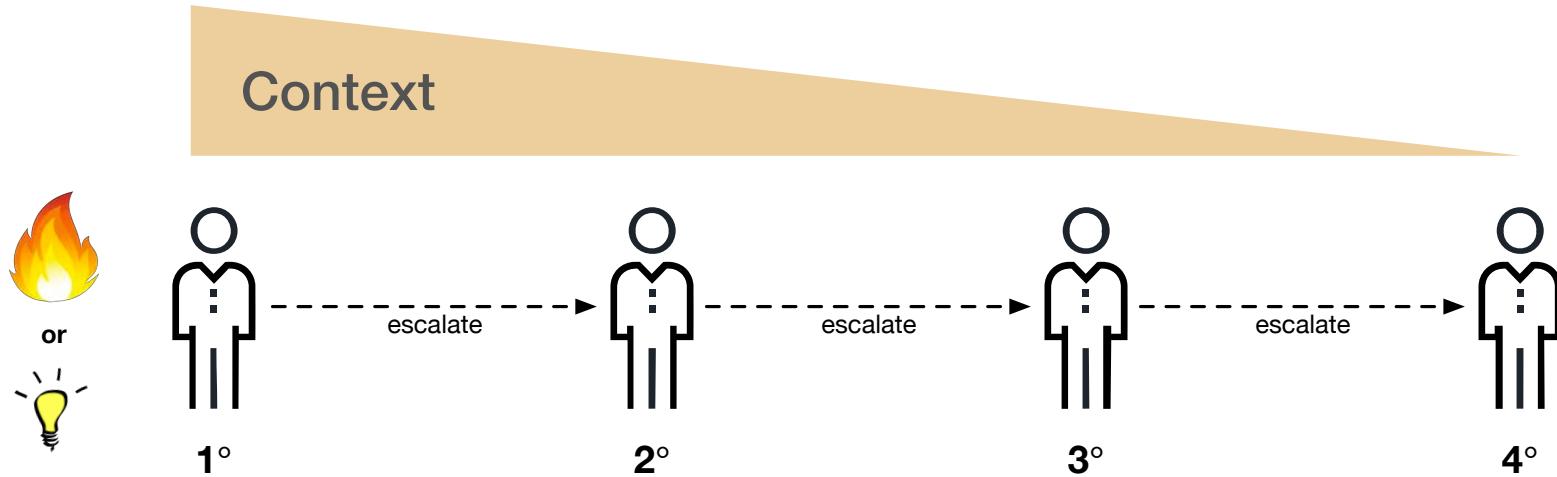
Answer is always
“it depends”



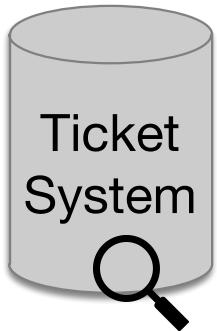
John
Allspaw

RUNDECK

Where are decisions made? Who can take action?

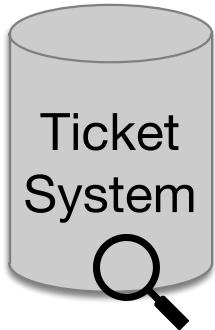


Low trust + approvals = illusion of control

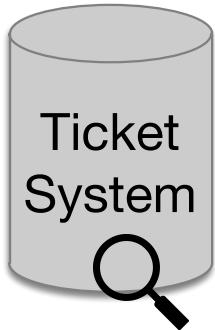


Low trust + approvals = illusion of control

Add up the total number of approval requests and

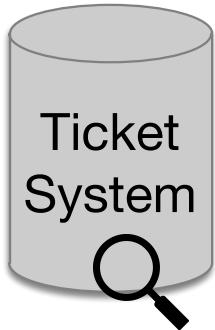


Low trust + approvals = illusion of control



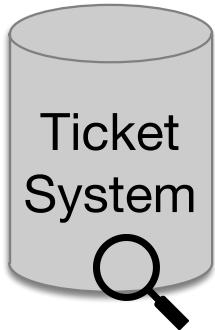
Add up the total number of approval requests and
...subtract the info radiators (“I need to be in the loop”)

Low trust + approvals = illusion of control



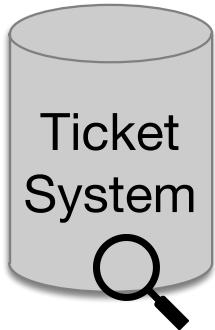
Add up the total number of approval requests and
...subtract the info radiators (“I need to be in the loop”)
...subtract the CYAs (“Prove you followed the process”)

Low trust + approvals = illusion of control



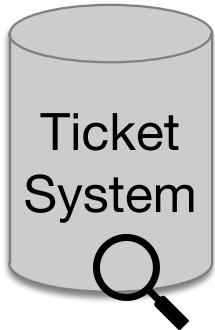
Add up the total number of approval requests and
...subtract the info radiators (“I need to be in the loop”)
...subtract the CYAs (“Prove you followed the process”)
...subtract the too removed to judge (“mostly guessing”)

Low trust + approvals = illusion of control



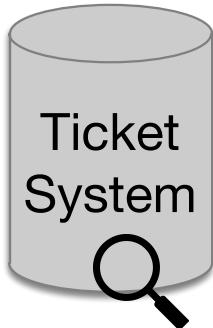
Add up the total number of approval requests and
...subtract the info radiators (“I need to be in the loop”)
...subtract the CYAs (“Prove you followed the process”)
...subtract the too removed to judge (“mostly guessing”)

Low trust + approvals = illusion of control



Add up the total number of approval requests and
...subtract the info radiators (“I need to be in the loop”)
...subtract the CYAs (“Prove you followed the process”)
...subtract the too removed to judge (“mostly guessing”)

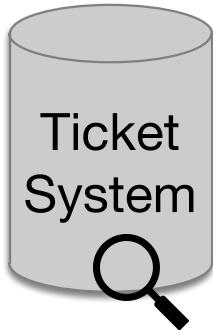
Low trust + approvals = illusion of control



Add up the total number of approval requests and
...subtract the info radiators (“I need to be in the loop”)
...subtract the CYAs (“Prove you followed the process”)
...subtract the too removed to judge (“mostly guessing”)

How many are you left with?

Low trust + approvals = illusion of control



Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

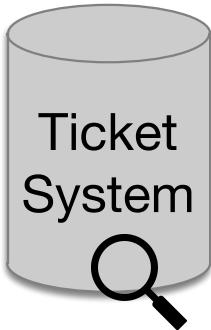
...subtract the CYAs (“Prove you followed the process”)

...subtract the too removed to judge (“mostly guessing”)

How many are you left with?

How many were the right call?

Low trust + approvals = illusion of control



Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

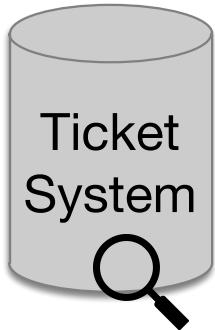
...subtract the too removed to judge (“mostly guessing”)

How many are you left with?

How many were the right call?

How many got rejected?

Low trust + approvals = illusion of control



Add up the total number of approval requests and

...subtract the info radiators (“I need to be in the loop”)

...subtract the CYAs (“Prove you followed the process”)

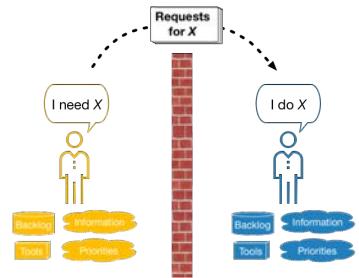
...subtract the too removed to judge (“mostly guessing”)

How many are you left with?

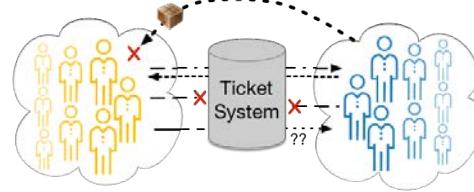
How many were the right call?

How many got rejected?

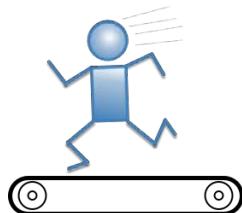
Forces That Undermine Operations



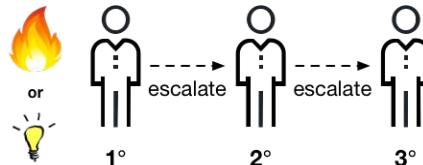
Silos



Queues



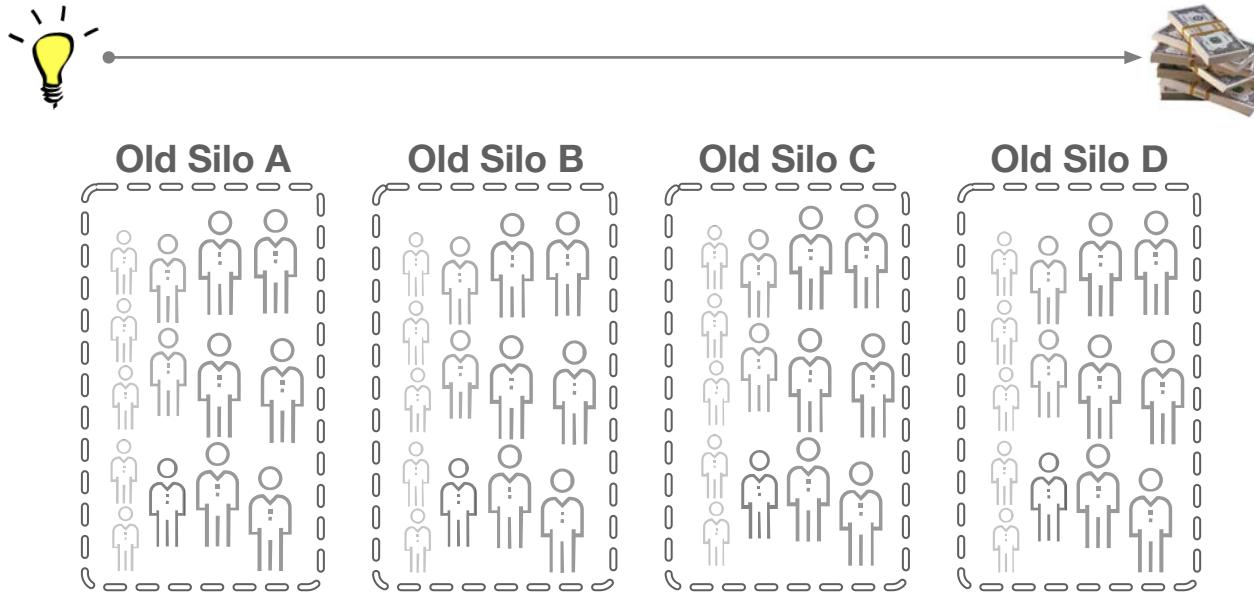
Toil



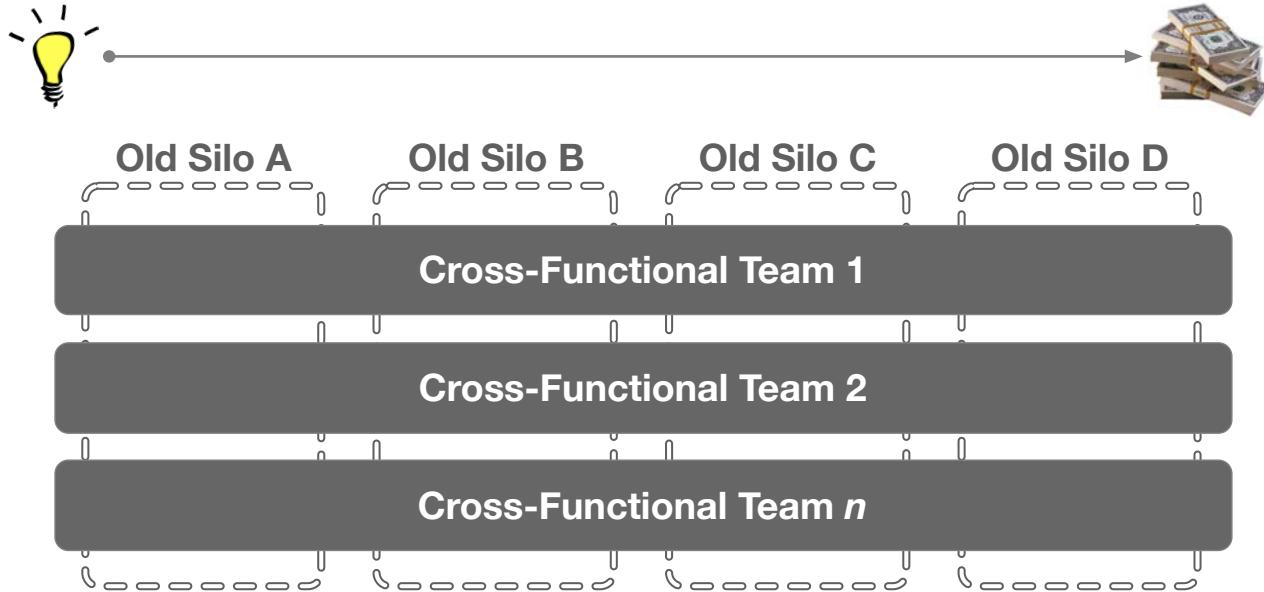
Low Trust

So what can we do differently?

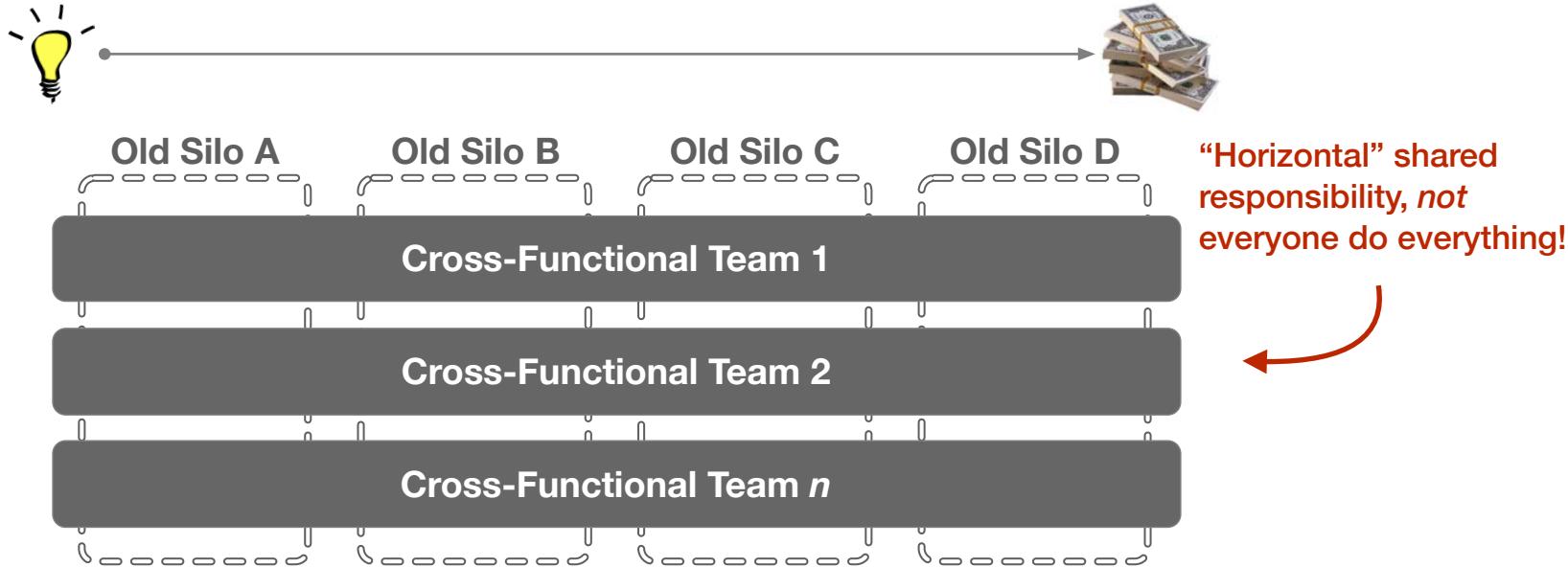
Obvious: Get rid of as many silos as possible



Obvious: Get rid of as many silos as possible

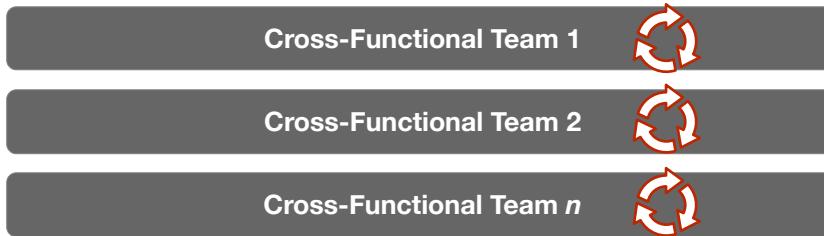


Obvious: Get rid of as many silos as possible



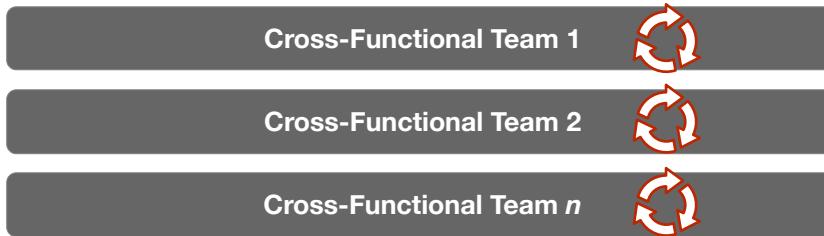
Shared and dedicated responsibility is key

*“Netflix”
Model*



Shared and dedicated responsibility is key

**“Netflix”
Model**



**“Google”
Model**

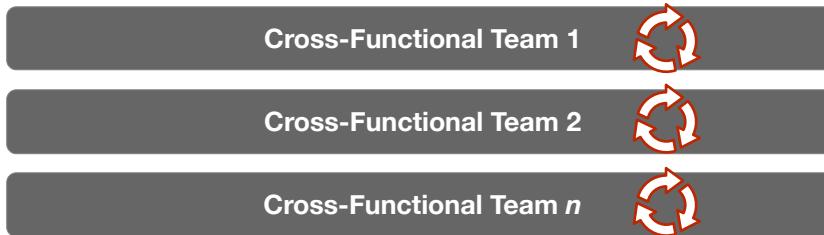


Clear handoff requirements →

← Error budget consequences

Shared and dedicated responsibility is key

*“Netflix”
Model*



*“Google”
Model*

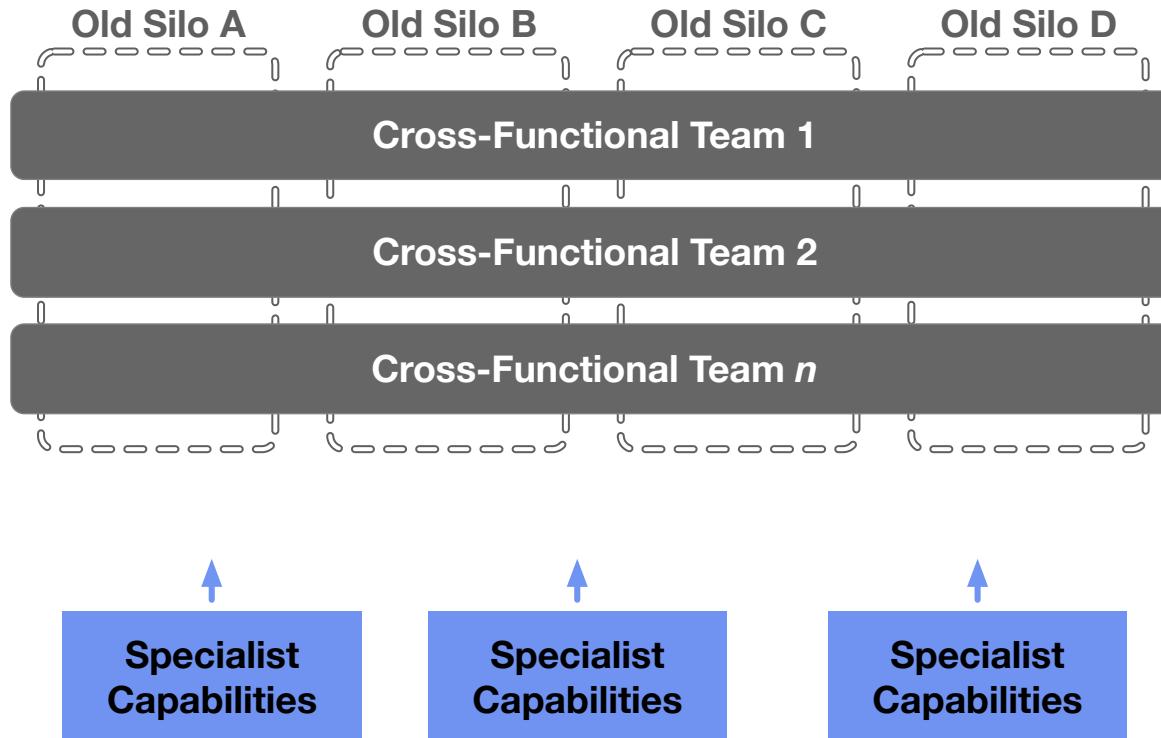


*Same
high-quality,
high-velocity
results!*

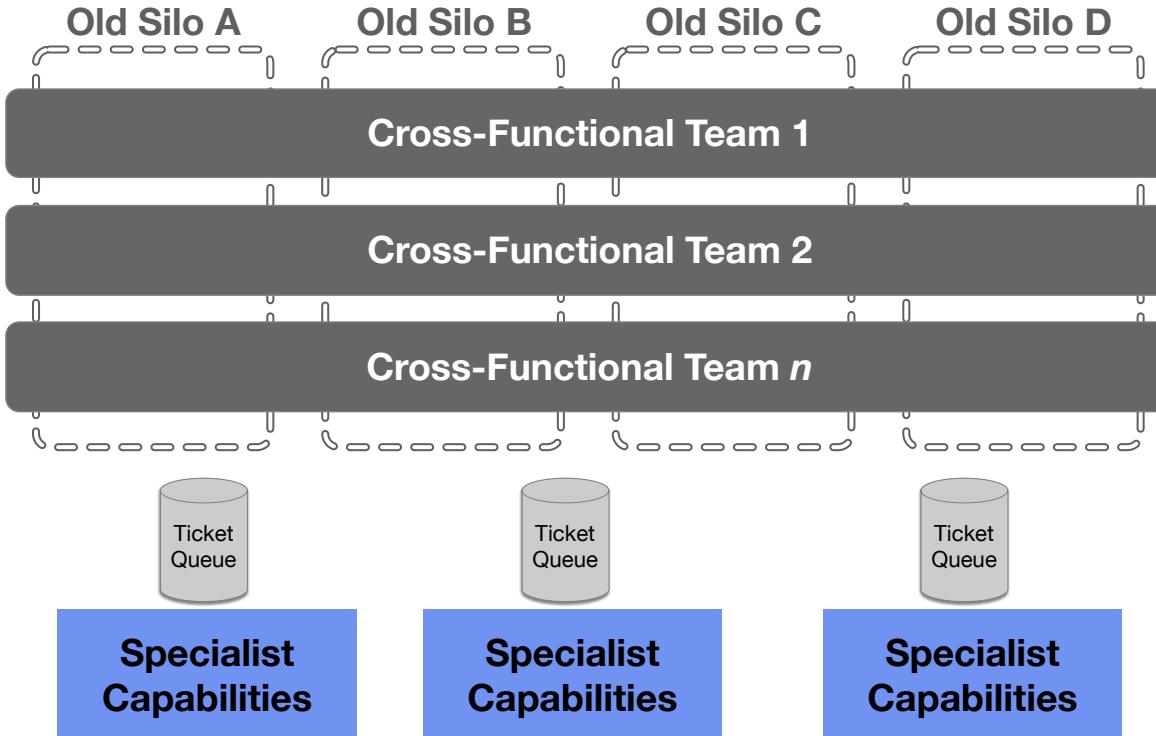
Clear handoff requirements

Error budget consequences

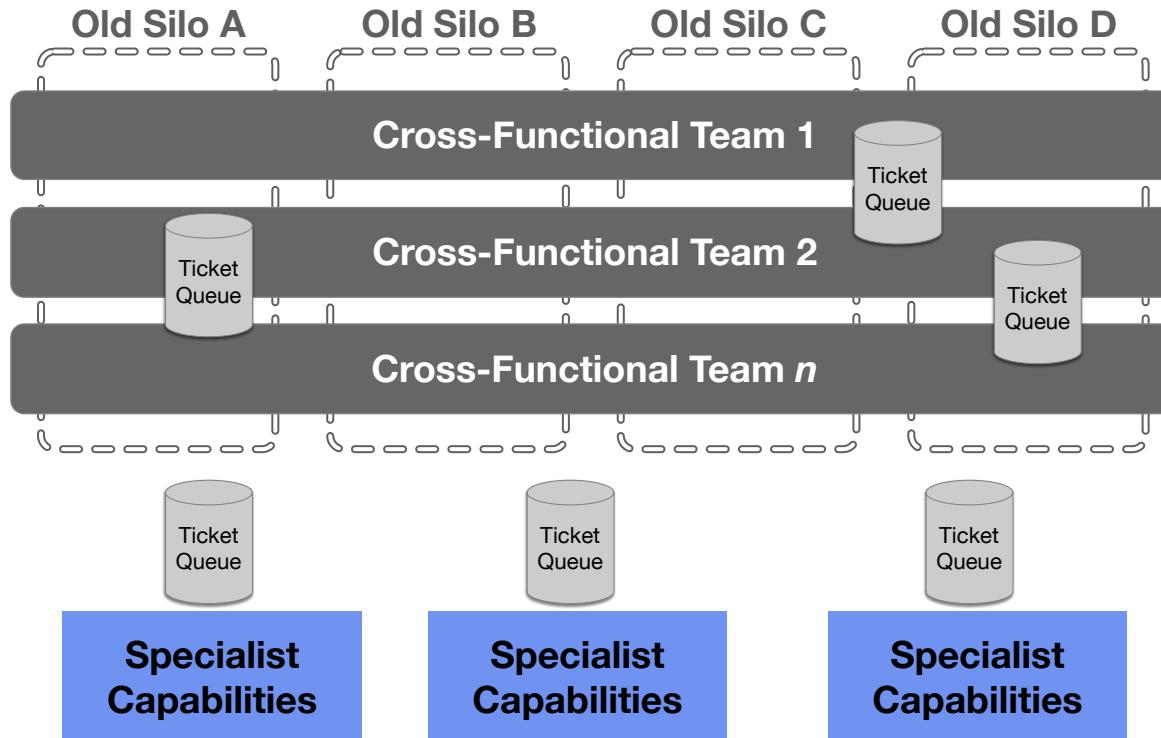
But what about the cross-cutting concerns?



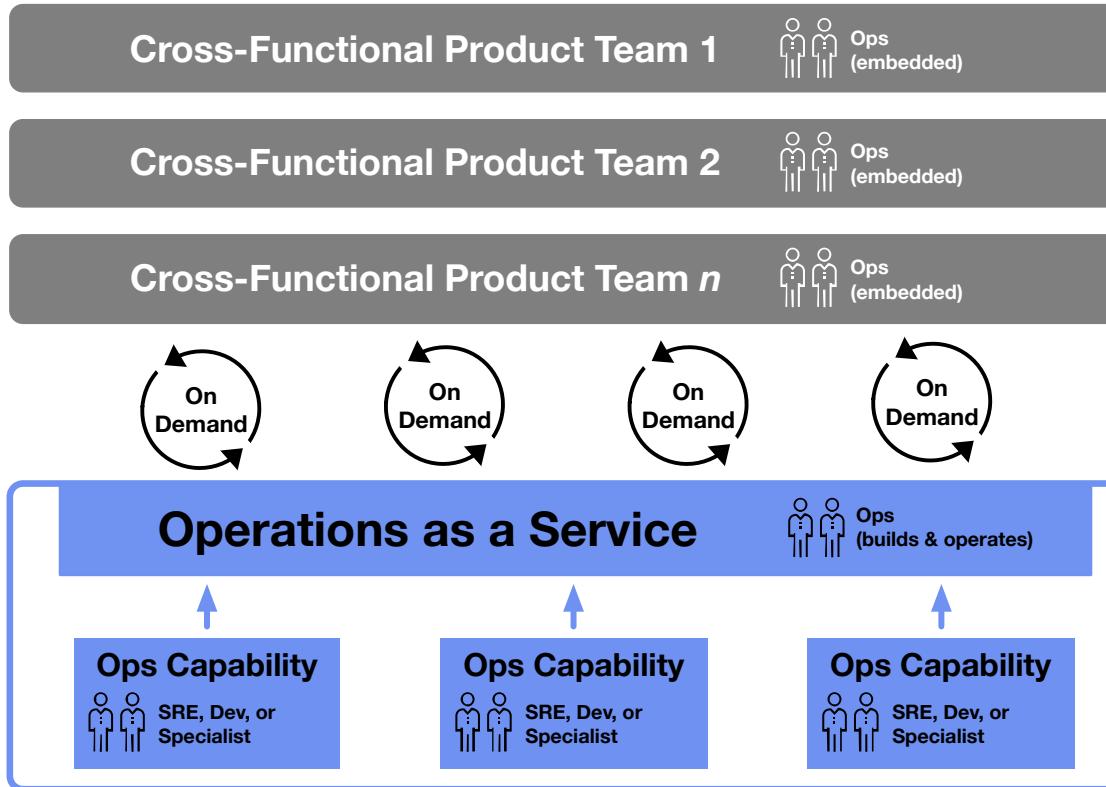
But what about the cross-cutting concerns?



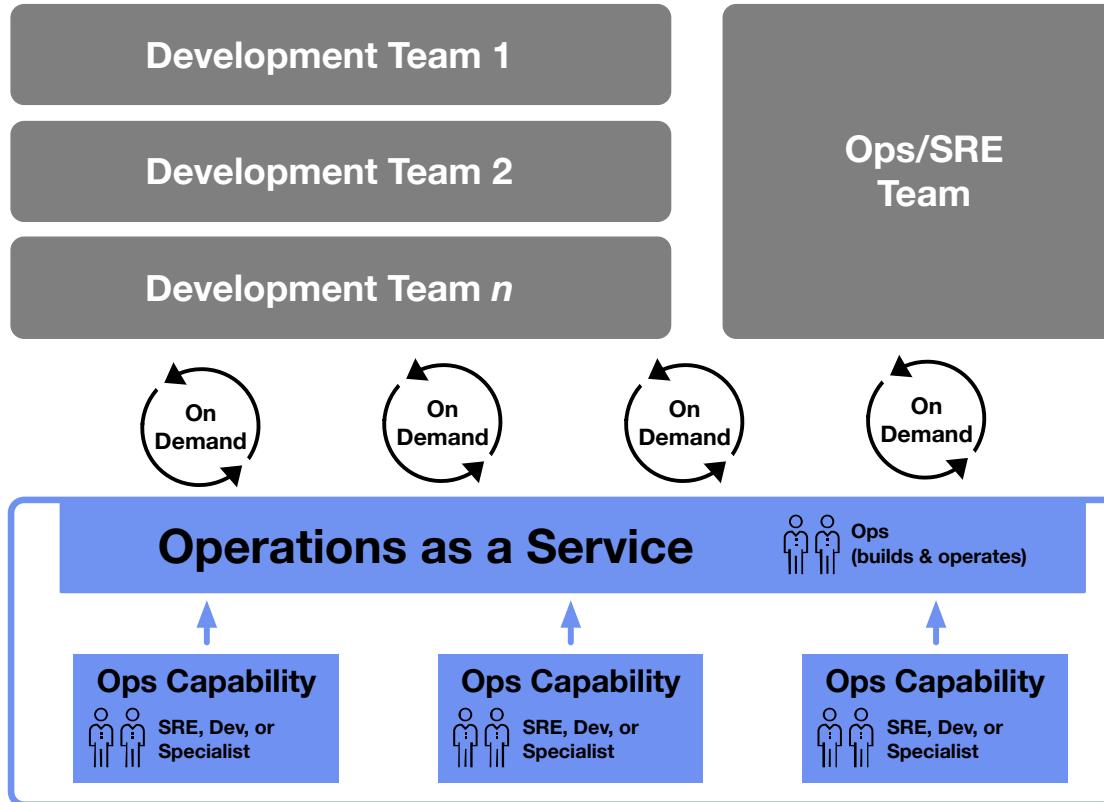
But what about the cross-cutting concerns?



Operations as a Service: Turn handoffs into self-service



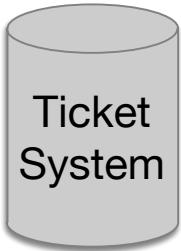
Operations as a Service: Works with any org model



Use tickets only for what they are good for

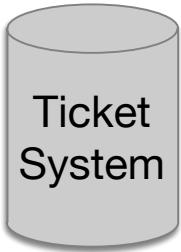


Use tickets only for what they are good for



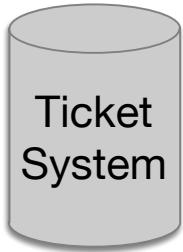
1. Documenting true problems/issues/exceptions

Use tickets only for what they are good for



1. Documenting true problems/issues/exceptions
2. Routing for necessary approvals

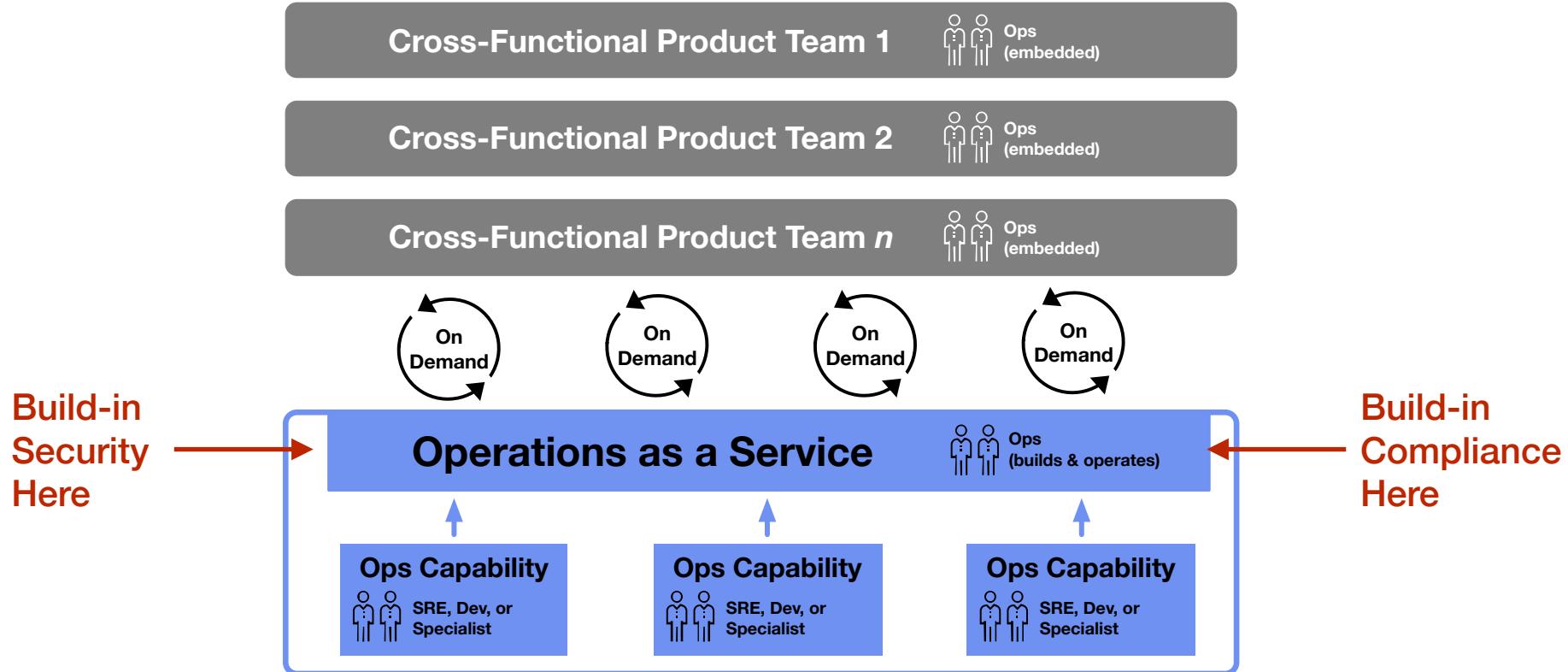
Use tickets only for what they are good for



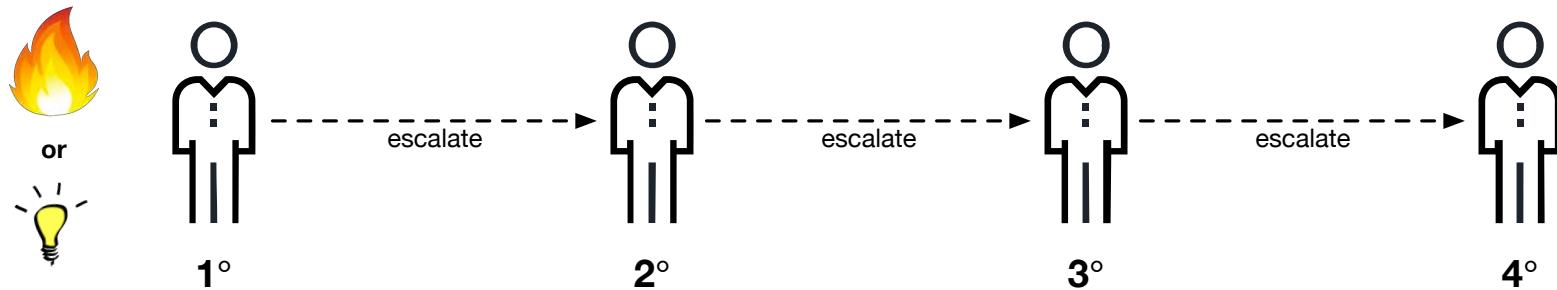
1. Documenting true problems/issues/exceptions
2. Routing for necessary approvals

Not as a general purpose work management system!

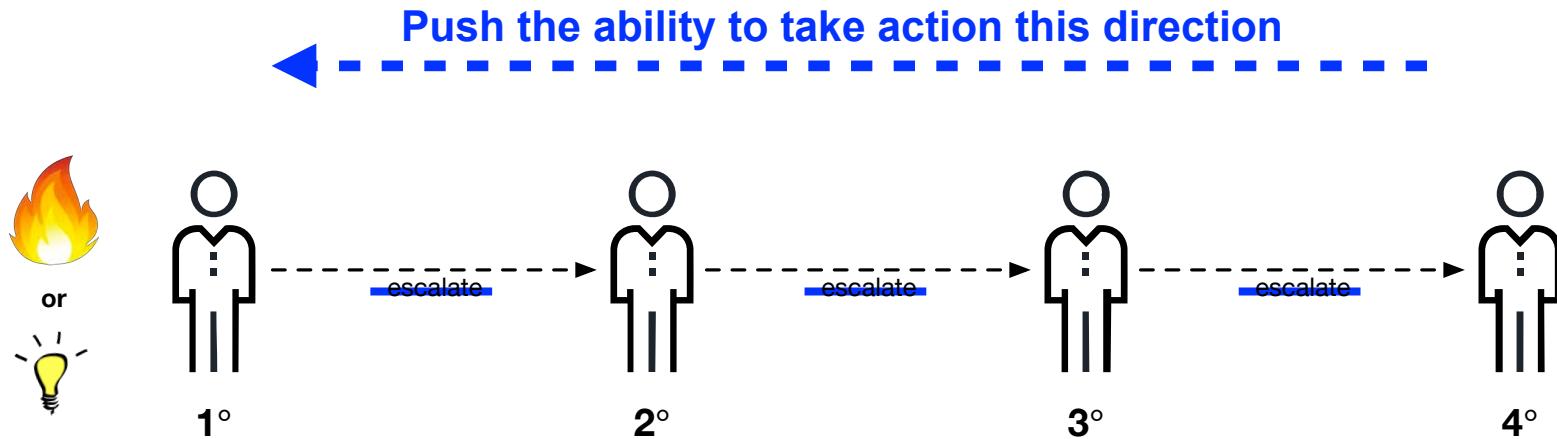
Security or compliance “in the way”?



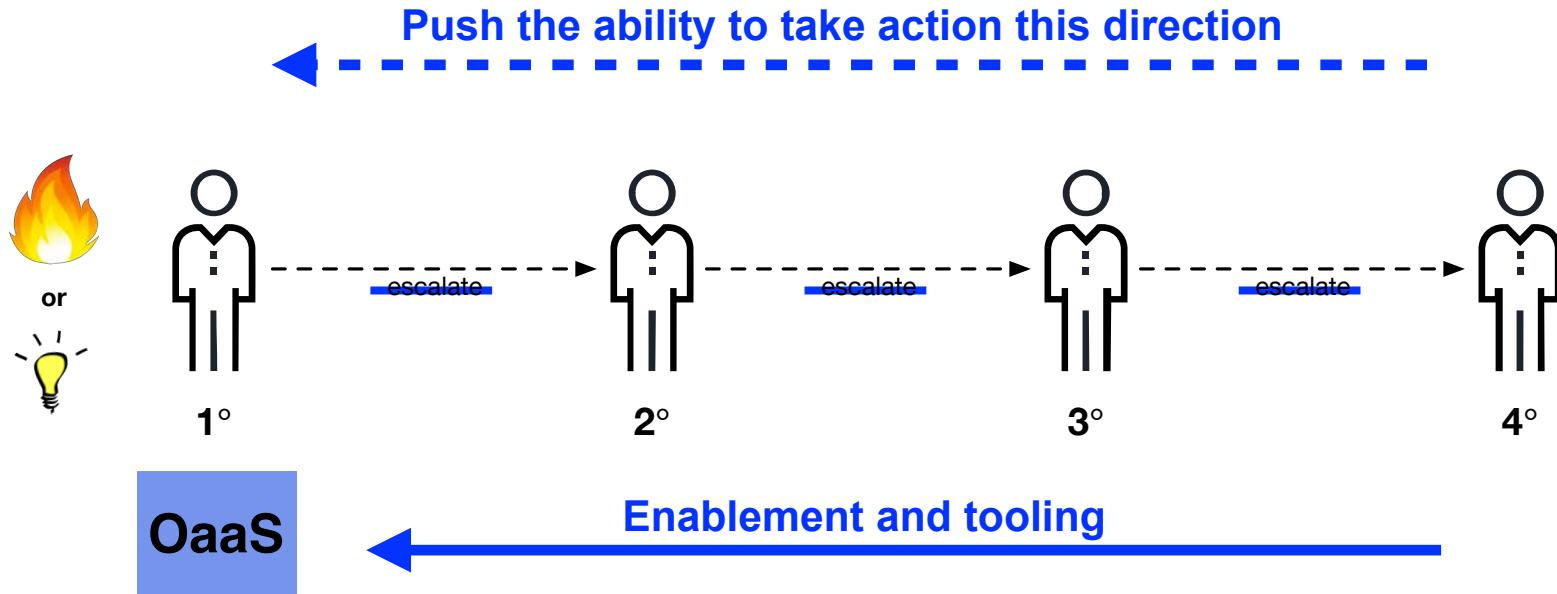
“Shift Left” the ability to take action



“Shift Left” the ability to take action



“Shift Left” the ability to take action



Reduce Toil

Reduce Toil

1. Track toil levels for each team

Reduce Toil

1. Track toil levels for each team
2. Set toil limits for each team

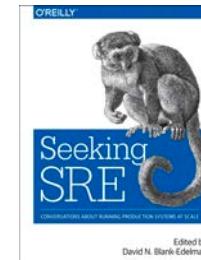
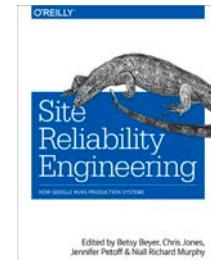
Reduce Toil

1. Track toil levels for each team
2. Set toil limits for each team
3. Fund efforts to reduce toil (with emphasis on teams over toil limits)

Reduce Toil

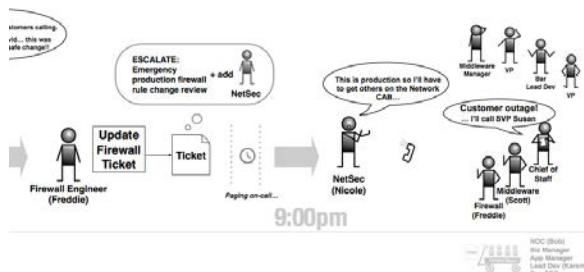
1. Track toil levels for each team
2. Set toil limits for each team
3. Fund efforts to reduce toil (with emphasis on teams over toil limits)

Bonus: Use Service Level Objectives, Error Budgets, and other lessons from SRE

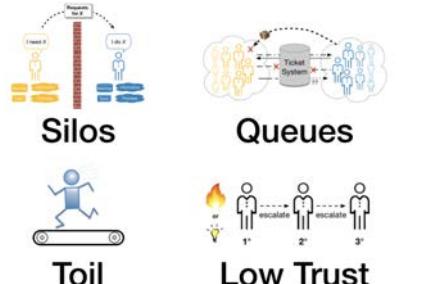


Recap

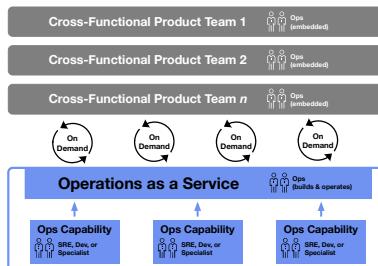
Don't forget about Ops.
Challenge conventional wisdom.



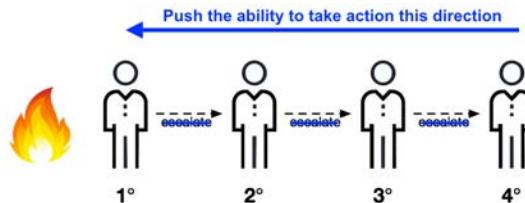
Understand the forces undermining operations work



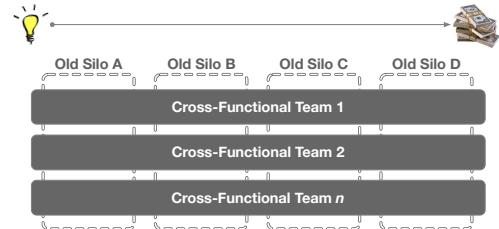
Leverage the Operations as a Service design pattern



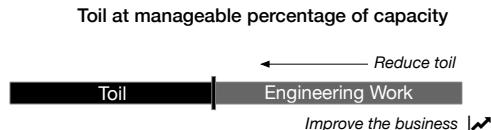
"Shift-Left" control and decision making.



Focus on removing silos and queues



Learn from SRE: Reduce toil to create capacity to change



Let's talk...



damon@rundeck.com



@damonedwards

Dive Deeper Into Operations as a Service:

<https://www.rundeck.com/oaas>