Failure Happens: Improving Incident Response In Enterprises

Damon Edwards
@damonedwards
Please note….

1. This is not a talk about monitoring

2. This talk is about the enterprise
Let’s look at an (unfortunately) typical incident...

[Get PDF here: https://rundeck.co/incident_lisa2017]
## Let’s condense that…

<table>
<thead>
<tr>
<th>Time</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td><strong>NOC alerts</strong></td>
</tr>
<tr>
<td>12:00pm</td>
<td><strong>L2 bridge call</strong></td>
</tr>
<tr>
<td>2:00pm</td>
<td><strong>Interrupt Foo app dev</strong></td>
</tr>
<tr>
<td>2:30pm</td>
<td><strong>Identify incorrect restarts</strong></td>
</tr>
<tr>
<td>5:00pm</td>
<td><strong>Escalate to SVP</strong></td>
</tr>
<tr>
<td>6:00pm</td>
<td><strong>Find middleware engineer</strong></td>
</tr>
<tr>
<td>9:30pm</td>
<td><strong>Need API tests per policy calls restart done</strong></td>
</tr>
<tr>
<td>10:30pm</td>
<td><strong>Middleware calls restart done</strong></td>
</tr>
</tbody>
</table>

### Actions:
- **NOC alerts**
- **Biz Manager phone call**
- **SysAdmin try this & that**
- **L2 bridge call**
- **Escalate to L2**
- **Consensus: Foo service**
- **Interrupt Foo with access**
- **Attempt to get middleware to restart**
- **Multiple roundtrips to get logs**
- **Find sysadmin with access**
- **Find middleware to restart**
- **Identify incorrect restarts**
- **Approve**
- **Review docs**
- **Find tester**
- **Pass tests**
- **Middleware calls restart done**
- **NOC sees green**
- **Ticket closed**

### Timeline:
- 10:00am: **NOC alerts**
- 12:00pm: **L2 bridge call**
- 2:00pm: **Interrupt Foo app dev**
- 2:30pm: **Identify incorrect restarts**
- 5:00pm: **Escalate to SVP**
- 6:00pm: **Find middleware engineer**
- 9:30pm: **Need API tests per policy**
- 10:30pm: **Middleware calls restart done**
Take a DevOps approach to improvement...
1. Use Lean thinking and analyze like any process
## What got in the way? (Wastes)

<table>
<thead>
<tr>
<th>Time</th>
<th>Action</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>NOC alerts</td>
<td>NOC (Bob) Business Manager</td>
</tr>
<tr>
<td>+00:30 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00pm</td>
<td>-L2 bridge call</td>
<td>NOC (Bob), Biz Manager, SysAdmin (Steve)</td>
</tr>
<tr>
<td>+02:30 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:00pm</td>
<td>-Interrupt Foo app dev</td>
<td>NOC (Bob), Biz Manager, Lead Dev (Karen), Foo L2, SysAdmin (Lee)</td>
</tr>
<tr>
<td>+04:30 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:30pm</td>
<td>-Identify incorrect restarts</td>
<td>NOC (Bob), Biz Manager, Lead Dev (Karen), Foo L2, SysAdmin (Lee)</td>
</tr>
<tr>
<td>+05:00 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:00pm</td>
<td>-Escalate to SVP</td>
<td>NOC (Bob), Biz Manager, Lead Dev (Karen), Foo L2, SysAdmin (Lee)</td>
</tr>
<tr>
<td>+07:30 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:00pm</td>
<td>-Find middleware engineer</td>
<td>NOC (Bob), Biz Manager, Lead Dev (Karen), Foo L2, SysAdmin (Lee)</td>
</tr>
<tr>
<td>+08:30 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30pm</td>
<td>-Need API tests per policy</td>
<td>NOC (Bob), Biz Manager, Lead Dev (Karen), Foo L2, SysAdmin (Lee)</td>
</tr>
<tr>
<td>+12:00 after alert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:30pm</td>
<td>-Middleware calls restart done</td>
<td>NOC (Bob), Biz Manager, Lead Dev (Karen), Foo L2, SysAdmin (Lee)</td>
</tr>
<tr>
<td>+13:00 after alert</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

- **PD** - Partially Done
- **TS** - Task Switching
- **W** - Waiting
- **M** - Motion / Manual
- **D** - Defects
- **EP** - Extra Process
- **EF** - Extra Features
- **H** - Heroics
<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>NOC alerts</td>
</tr>
<tr>
<td>12:00pm</td>
<td>L2 bridge call</td>
</tr>
<tr>
<td>2:00pm</td>
<td>L2 bridge call</td>
</tr>
<tr>
<td>2:30pm</td>
<td>Interrupt Foo app dev</td>
</tr>
<tr>
<td></td>
<td>SysAdmin try this &amp; that</td>
</tr>
<tr>
<td></td>
<td>Consensus: Foo service</td>
</tr>
<tr>
<td></td>
<td>Multiple roundtrips to get logs</td>
</tr>
<tr>
<td></td>
<td>Identify incorrect restarts</td>
</tr>
<tr>
<td></td>
<td>Attempt to get middleware to restart</td>
</tr>
<tr>
<td></td>
<td>Need approval</td>
</tr>
<tr>
<td>5:00pm</td>
<td>Escalate to SVP</td>
</tr>
<tr>
<td></td>
<td>VPs try to assess impact</td>
</tr>
<tr>
<td></td>
<td>Approve</td>
</tr>
<tr>
<td>6:00pm</td>
<td>Find middleware engineer</td>
</tr>
<tr>
<td></td>
<td>Review docs</td>
</tr>
<tr>
<td></td>
<td>Trial &amp; error</td>
</tr>
<tr>
<td>9:30pm</td>
<td>Need API tests per policy</td>
</tr>
<tr>
<td>10:30pm</td>
<td>Middleware calls restart done</td>
</tr>
<tr>
<td></td>
<td>-NOCC sees</td>
</tr>
<tr>
<td></td>
<td>-Ticket closed</td>
</tr>
</tbody>
</table>

Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

- **PD** - Partially Done
- **TS** - Task Switching
- **W** - Waiting
- **M** - Motion / Manual
- **D** - Defects
- **EP** - Extra Process
- **EF** - Extra Features
- **H** - Heroics
Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

- **PD** - Partially Done
- **TS** - Task Switching
- **W** - Waiting
- **M** - Motion / Manual
- **D** - Defects
- **EP** - Extra Process
- **EF** - Extra Features
- **H** - Heroics
What got in the way? (Wastes)

1. -NOC alerts
   -Biz Manager phone call
   -Escalate to L2

2. -L2 bridge call
   -SysAdmin try this & that
   -Consensus: Foo service

3. -Interrupt Foo app dev
   -Find sysadmin with access
   -Multiple roundtrips to get logs
   -Identify incorrect restarts

4. -Attempt to get middleware to restart
   -Need approval

5. -Escalate to SVP
   -VPs try to assess impact
   -Approve

6. -Find middleware engineer
   -Review docs
   -Trial & error
   -Need API tests per policy
   -Find tester
   -Pass tests

7. -Middleware calls restart done
   -NOC sees green
   -Ticket closed

8. Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:
   
   PD - Partially Done
   TS - Task Switching
   W - Waiting
   M - Motion / Manual
   D - Defects
   EP - Extra Process
   EF - Extra Features
   H - Heroics
Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

- **PD** - Partially Done
- **TS** - Task Switching
- **W** - Waiting
- **M** - Motion / Manual
- **D** - Defects
- **EP** - Extra Process
- **EF** - Extra Features
- **H** - Heroics
Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Partially Done</td>
<td>Task Switching</td>
<td>Waiting</td>
<td>Motion / Manual</td>
<td>Extra Process</td>
<td>Task Switching</td>
<td>Motion / Manual</td>
<td>Waiting</td>
<td>Motion / Manual</td>
<td>Task Switching</td>
<td>Task Switching</td>
<td>Extra Process</td>
<td>Waiting</td>
<td>Motion / Manual</td>
<td>Waiting</td>
<td></td>
</tr>
</tbody>
</table>

- **PD** - Partially Done
- **TS** - Task Switching
- **W** - Waiting
- **M** - Motion / Manual
- **EP** - Extra Process
- **EF** - Extra Features
- **H** - Heroics
What got in the way? (Wastes)

Structured analysis building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

PD - Partially Done
TS - Task Switching
W - Waiting
M - Motion / Manual
D - Defects
EP - Extra Process
EF - Extra Features
H - Heroics
Tip: Try techniques like Value Stream Mapping
1. Use Lean thinking and analyze like any process
2. Make sure you are measuring the right things
**MTTD?**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-NOC alerts</td>
<td>-L2 bridge call</td>
<td>-Interrupt Foo app dev</td>
<td>-Identify incorrect restarts</td>
<td>-Escalate to SVP</td>
<td>-Find middleware engineer</td>
<td>-Need API tests per policy</td>
<td>-Middleware calls restart done</td>
</tr>
<tr>
<td></td>
<td>-Biz Manager</td>
<td>-SysAdmin try this &amp; that</td>
<td>-Find sysadmin with access</td>
<td>-Attempt to get middleware to restart</td>
<td>-Approve</td>
<td>-Review docs</td>
<td>-Find tester</td>
<td>-NOC sees green</td>
</tr>
<tr>
<td></td>
<td>phone call</td>
<td>-Consensus: Foo service</td>
<td>-Multiple roundtrips to get logs</td>
<td>-Need approval</td>
<td></td>
<td></td>
<td>-Pass tests</td>
<td>-Ticket closed</td>
</tr>
<tr>
<td>10:00am</td>
<td>12:00pm</td>
<td>2:00pm</td>
<td>2:30pm</td>
<td>5:00pm</td>
<td>6:00pm</td>
<td>9:30pm</td>
<td>10:30pm</td>
<td></td>
</tr>
<tr>
<td>+00:30 after alert</td>
<td>+02:30 after alert</td>
<td>+04:30 after alert</td>
<td>+05:00 after alert</td>
<td>+07:30 after alert</td>
<td>+08:30 after alert</td>
<td>+12:00 after alert</td>
<td>+13:00 after alert</td>
<td></td>
</tr>
</tbody>
</table>

NOC (Bob) Biz Manager App Manager Lead Dev (Karen) Foo L2 SysAdmin (Lee) Middleware Mngr. SVP Chief of Staff 2 x Tech VP Cust. Engage. (Varsha)
### MTTD?

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td><strong>2</strong></td>
<td><strong>3</strong></td>
<td><strong>4</strong></td>
<td><strong>5</strong></td>
<td><strong>6</strong></td>
<td><strong>7</strong></td>
<td><strong>8</strong></td>
</tr>
<tr>
<td>NOC alerts</td>
<td>L2 bridge call</td>
<td>-Interrupt Foo app dev</td>
<td>-Identify incorrect restarts</td>
<td>-Escalate to SVP</td>
<td>-Find middleware engineer</td>
<td>-Need API tests per policy</td>
<td>-Middleware calls restart done</td>
</tr>
<tr>
<td>Biz Manager phone call</td>
<td>-SysAdmin try this &amp; that</td>
<td>-Find sysadmin with access</td>
<td>-Attempt to get middleware to restart</td>
<td>-VPs try to assess impact</td>
<td>-Review docs</td>
<td>-Find tester</td>
<td>-NOC sees green</td>
</tr>
<tr>
<td>Escalate to L2</td>
<td>-Consensus: Foo service</td>
<td>-Multiple roundtrips to get logs</td>
<td>-Approve</td>
<td>-Trial &amp; error</td>
<td></td>
<td>-Pass tests</td>
<td>-Ticket closed</td>
</tr>
</tbody>
</table>

**10:00am**
- NOC (Bob)
- Biz Manager
- SysAdmin (Steve)

**12:00pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**2:00pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**2:30pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**5:00pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**6:00pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**9:30pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**10:30pm**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.

**Detect?**

- NOC (Bob)
- Biz Manager
- SysAdmin (Steve)
- 7 x L2

- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.
- SVP

- Chief of Staff
- 2 x Tech VP
- Middleware (Scott)
- Cust. Engage. (Varsha)
MTTD?

1. -NOC alerts
   -Biz Manager phone call
   -Escalate to L2

2. -L2 bridge call
   -SysAdmin try this & that
   -Consensus: Foo service

3. -Interrupt Foo app dev
   -Find sysadmin with access
   -Multiple roundtrips to get logs

4. -Identify incorrect restarts
   -Attempt to get middleware to restart
   -Need approval
   -Escalate to SVP

5. -Find middleware engineer
   -VPs try to assess impact
   -Approve
   -Review docs
   -Trial & error

6. -Need API tests per policy
   -Find tester
   -Pass tests

7. -Middleware calls restart done
   -NOC sees green
   -Ticket closed

8. -Detect?

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

   Detect?

   Diagnose!

  Detect?
<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>-NOCT alerts</td>
</tr>
<tr>
<td>12:00pm</td>
<td>-SysAdmin try app dev</td>
</tr>
<tr>
<td>2:00pm</td>
<td>-Interrupt Foo app dev</td>
</tr>
<tr>
<td>2:30pm</td>
<td>-Identify incorrect restarts</td>
</tr>
<tr>
<td>5:00pm</td>
<td>-Escalate to SVP</td>
</tr>
<tr>
<td>6:00pm</td>
<td>-Find middleware engineer</td>
</tr>
<tr>
<td>9:30pm</td>
<td>-Need approval</td>
</tr>
<tr>
<td>10:30pm</td>
<td>-Middleware calls restart done</td>
</tr>
</tbody>
</table>

**Diagnose > Detect**

<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00pm</td>
<td>-NOC sees green</td>
</tr>
<tr>
<td>1:30pm</td>
<td>-Ticket closed</td>
</tr>
</tbody>
</table>

**Detect?**

1. NOC (Bob)
   Biz Manager
   SysAdmin (Steve)
   7 x L2

2. NOC (Bob)
   Biz Manager
   App Manager
   Lead Dev (Karen)
   Foo L2
   SysAdmin (Lee)

3. NOC (Bob)
   Biz Manager
   App Manager
   Lead Dev (Karen)
   Foo L2

4. NOC (Bob)
   Biz Manager
   App Manager
   Lead Dev (Karen)
   Foo L2
   SysAdmin (Lee)
   Middleware Mngr.
   SVP
   Chief of Staff
   2 x Tech VP
   Middleware (Scott)
   Cust. Engage.
   (Varsha)

5. NOC (Bob)
   Biz Manager
   App Manager
   Lead Dev (Karen)
   Foo L2
   SysAdmin (Lee)
   Middleware Mngr.
   SVP
   Chief of Staff
   2 x Tech VP
   Middleware (Scott)
   Cust. Engage.
   (Varsha)
<table>
<thead>
<tr>
<th>Time</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>- NOC alerts</td>
</tr>
<tr>
<td>+00:30</td>
<td>after alert</td>
</tr>
<tr>
<td>12:00pm</td>
<td>- L2 bridge call</td>
</tr>
<tr>
<td>+02:30</td>
<td>after alert</td>
</tr>
<tr>
<td>2:00pm</td>
<td>- Interrupt Foo app dev</td>
</tr>
<tr>
<td>+04:30</td>
<td>after alert</td>
</tr>
<tr>
<td>2:30pm</td>
<td>- Find sysadmin with access</td>
</tr>
<tr>
<td>+05:00</td>
<td>after alert</td>
</tr>
<tr>
<td>5:00pm</td>
<td>- Identify incorrect restarts</td>
</tr>
<tr>
<td></td>
<td>- Attempt to get middleware to restart</td>
</tr>
<tr>
<td></td>
<td>- Need approval</td>
</tr>
<tr>
<td>6:00pm</td>
<td>- Escalate to SVP</td>
</tr>
<tr>
<td></td>
<td>- VPs try to assess impact</td>
</tr>
<tr>
<td></td>
<td>- Review docs</td>
</tr>
<tr>
<td></td>
<td>- Trial &amp; error</td>
</tr>
<tr>
<td>9:30pm</td>
<td>- Need API tests per policy</td>
</tr>
<tr>
<td>+12:00</td>
<td>after alert</td>
</tr>
<tr>
<td>10:30pm</td>
<td>- Middleware calls restart done</td>
</tr>
<tr>
<td>+13:00</td>
<td>after alert</td>
</tr>
</tbody>
</table>

**NOC (Bob)**
- Biz Manager
- SysAdmin (Steve)
- Lead Dev (Karen)
- Foo L2
- Middleware Mngr.
- SVP
- Chief of Staff
- Cust. Engage.

**Biz Manager**
- Call manager
- Systems Admin
- Call lead dev
- Call middleware
- Call SVP
- Call Chief of Staff
- Call Cust. Engage.

**SysAdmin (Steve)**
- Call lead dev
- Call middleware
- Call SVP
- Call Chief of Staff
- Call Cust. Engage.

**Lead Dev (Karen)**
- Call middleware
- Call SVP
- Call Chief of Staff
- Call Cust. Engage.

**Foo L2**
- Call lead dev
- Call middleware
- Call SVP
- Call Chief of Staff
- Call Cust. Engage.

**Middleware Mngr.**
- Call SVP
- Call Chief of Staff
- Call Cust. Engage.

**SVP**
- Play leadership role
- Call Chief of Staff
- Call Cust. Engage.

**Chief of Staff**
- Call Cust. Engage.

**2 x Tech VP**
- Call Cust. Engage.

**Middleware (Scott)**
- Call Cust. Engage.
## MTTR?

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>- NOC alerts</td>
</tr>
<tr>
<td>2</td>
<td>- Biz Manager phone call</td>
</tr>
<tr>
<td>3</td>
<td>- Escalate to L2</td>
</tr>
<tr>
<td>4</td>
<td>- Interrupt Foo app dev</td>
</tr>
<tr>
<td>5</td>
<td>- Identify incorrect restarts</td>
</tr>
<tr>
<td>6</td>
<td>- Escalate to SVP</td>
</tr>
<tr>
<td>7</td>
<td>- Find middleware engineer</td>
</tr>
<tr>
<td>8</td>
<td>- Middleware calls restart done</td>
</tr>
</tbody>
</table>

### Timeline

<table>
<thead>
<tr>
<th>Time</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>NOC (Bob) Biz Manager</td>
</tr>
<tr>
<td>12:00pm</td>
<td>NOC (Bob) Biz Manager SysAdmin (Steve) 7 x L2</td>
</tr>
<tr>
<td>2:00pm</td>
<td>NOC (Bob) Biz Manager App Manager Lead Dev (Karen) Foo L2 SysAdmin (Lee) Middleware Mngr.</td>
</tr>
<tr>
<td>2:30pm</td>
<td>NOC (Bob) Biz Manager App Manager Lead Dev (Karen) Foo L2 SysAdmin (Lee) Middleware Mngr. SVP Chief of Staff 2 x Tech VP</td>
</tr>
<tr>
<td>5:00pm</td>
<td>NOC (Bob) Biz Manager App Manager Lead Dev (Karen) Foo L2 SysAdmin (Lee) Middleware Mngr. SVP Chief of Staff 2 x Tech VP Middleware (Scott) Cust. Engage. (Varsha)</td>
</tr>
<tr>
<td>6:00pm</td>
<td>NOC (Bob) Biz Manager App Manager Lead Dev (Karen) Foo L2 SysAdmin (Lee) Middleware Mngr. SVP Chief of Staff 2 x Tech VP Middleware (Scott) Cust. Engage. (Varsha)</td>
</tr>
<tr>
<td>9:30pm</td>
<td>NOC (Bob) Biz Manager App Manager Lead Dev (Karen) Foo L2 SysAdmin (Lee) Middleware Mngr. SVP Chief of Staff 2 x Tech VP Middleware (Scott) Cust. Engage. (Varsha)</td>
</tr>
</tbody>
</table>
| 10:00pm | - Restore? -

### Additional

- Find sysadmin with access
- Multiple roundtrips to get logs
- Attempt to get middleware to restart
- Need approval
- VPs try to assess impact
- Approve
- Trial & error
- Need API tests per policy
- Find tester
- Pass tests
- Middleware calls restart done
- NOC sees green
- Ticket closed
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-NOC alerts</td>
<td>-L2 bridge call</td>
<td>-Interrupt Foo app dev</td>
<td>-Identify incorrect restarts</td>
<td>-Escalate to SVP</td>
<td>-Need API tests per policy</td>
<td>-Middleware calls restart done</td>
</tr>
<tr>
<td>2</td>
<td>-Biz Manager phone call</td>
<td>-SysAdmin try this &amp; that</td>
<td>-Find sysadmin with access</td>
<td>-Attempt to get middleware to restart</td>
<td>-Find middleware engineer</td>
<td>-Find tester</td>
<td>-NOC sees green</td>
</tr>
<tr>
<td>3</td>
<td>-Escalate to L2</td>
<td>-Consensus: Foo service</td>
<td>-Multiple roundtrips to get logs</td>
<td>-Approve</td>
<td>-Review docs</td>
<td>-Trial &amp; error</td>
<td>-Ticket closed</td>
</tr>
</tbody>
</table>

**MTTR?**

- NOC alerts
- Biz Manager phone call
- Escalate to L2

**-NOC alerts**
- L2 bridge call
- SysAdmin try this & that
- Consensus: Foo service

**-Biz Manager phone call**
- Intermittent Foo app dev
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Escalate to L2**
- Identify incorrect restarts
- Attempt to get middleware to restart
- Need approval

**-Consensus: Foo service**
- Interrupt Foo app dev
- Find sysadmin with access

**-Multiple roundtrips to get logs**
- Identify incorrect restarts
- Attempt to get middleware to restart
- Need approval

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Attempt to get middleware to restart**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Need approval**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access

**-Identify incorrect restarts**
- Find sysadmin with access
- Multiple roundtrips to get logs

**-Find sysadmin with access**
- Multiple roundtrips to get logs

**-Multiple roundtrips to get logs**
- Find sysadmin with access
MTTR?

Restore is good. Repair (fix) is ultimate measure.

1. NOC alerts
   - Biz Manager phone call
   - Escalate to L2

2. L2 bridge call
   - SysAdmin try this & that
   - Consensus: Foo service

3. Interrupt Foo app dev
   - Find sysadmin with access
   - Multiple roundtrips to get logs

4. Identify incorrect restarts
   - Attempt to get middleware to restart
   - Need approval

5. Escalate to SVP
   - VPs try to assess impact
   - Approve

6. Find middleware engineer
   - Review docs
   - Trial & error

7. Need API tests per policy
   - Find tester
   - Pass tests

8. Middleware calls restart done
   - NOC sees green
   - Ticket closed

10:00am +00:30 after alert
12:00pm +02:30 after alert
2:00pm +04:30 after alert
2:30pm +05:00 after alert
5:00pm +07:30 after alert
6:00pm +08:30 after alert
9:30pm +12:00 after alert
10:30pm +13:00 after alert
<table>
<thead>
<tr>
<th>Cost?</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-NOC alerts</td>
<td>-L2 bridge call</td>
<td>-Interrupt Foo app dev</td>
<td>-Identify incorrect restarts</td>
<td>-Escalate to SVP</td>
<td>-Find middleware engineer</td>
<td>-Need API tests per policy</td>
<td>-Middleware calls restart done</td>
</tr>
<tr>
<td>2</td>
<td>-Biz Manager phone call</td>
<td>-SysAdmin try this &amp; that</td>
<td>-Find sysadmin with access</td>
<td>-Attempt to get middleware to restart</td>
<td>-VPs try to assess impact</td>
<td>-Review docs</td>
<td>-Find tester</td>
<td>-NOC sees green</td>
</tr>
<tr>
<td>3</td>
<td>-Escalate to L2</td>
<td>-Consensus: Foo service</td>
<td>-Multiple roundtrips to get logs</td>
<td>-Approve</td>
<td>-Approve</td>
<td>-Trial &amp; error</td>
<td>-Pass tests</td>
<td>-Ticket closed</td>
</tr>
<tr>
<td>4</td>
<td>-Consensus: Foo service</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Timeline:**

- **10:00am**
  - +00:30 after alert
  - HC: 2
  - NOC (Bob)

- **12:00pm**
  - +02:30 after alert
  - HC: 9
  - Biz Manager
  - NOC (Bob)
  - Biz Manager
  - SysAdmin (Steve)
  - 7 x L2

- **2:00pm**
  - +04:30 after alert
  - HC: 2
  - NOC (Bob)
  - Biz Manager
  - App Manager
  - Lead Dev (Karen)
  - Foo L2
  - SysAdmin (Lee)

- **2:30pm**
  - +05:00 after alert
  - HC: 3
  - NOC (Bob)
  - Biz Manager
  - App Manager
  - Lead Dev (Karen)
  - Foo L2
  - SysAdmin (Lee)
  - Middleware Mngr.

- **5:00pm**
  - +07:30 after alert
  - HC: 4
  - NOC (Bob)
  - Biz Manager
  - App Manager
  - Lead Dev (Karen)
  - Foo L2
  - SysAdmin (Lee)
  - Middleware Mngr.
  - SVP
  - Chief of Staff
  - 2 x Tech VP

- **6:00pm**
  - +08:30 after alert
  - HC: 1.5
  - NOC (Bob)
  - Biz Manager
  - App Manager
  - Lead Dev (Karen)
  - Foo L2
  - SysAdmin (Lee)
  - Middleware Mngr.
  - SVP
  - Chief of Staff
  - 2 x Tech VP
  - Middleware (Scott)
  - Cust. Engage. (Varsha)

- **9:30pm**
  - +12:00 after alert
  - HC: 1.5
  - NOC (Bob)
  - Biz Manager
  - App Manager
  - Lead Dev (Karen)
  - Foo L2
  - SysAdmin (Lee)
  - Middleware Mngr.
  - SVP
  - Chief of Staff
  - 2 x Tech VP
  - Middleware (Scott)
  - Cust. Engage. (Varsha)

- **10:30pm**
  - +13:00 after alert
  - HC: 1.5
  - NOC (Bob)
  - Biz Manager
  - App Manager
  - Lead Dev (Karen)
  - Foo L2
  - SysAdmin (Lee)
  - Middleware Mngr.
  - SVP
  - Chief of Staff
  - 2 x Tech VP
  - Middleware (Scott)
  - Cust. Engage. (Varsha)

**Legend:**
- HC: 2
- HC: 9
- HC: 2
- HC: 3
- HC: 4
- HC: 1.5
- HC: 1.5
- HC: 1.5

**Participants:**
- NOC (Bob)
- Biz Manager
- App Manager
- Lead Dev (Karen)
- Foo L2
- SysAdmin (Lee)
- Middleware Mngr.
- SVP
- Chief of Staff
- 2 x Tech VP
- Middleware (Scott)
- Cust. Engage. (Varsha)
<table>
<thead>
<tr>
<th>Time</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>- NOC alerts, Biz Manager phone call, Escalate to L2</td>
</tr>
<tr>
<td>12:00pm</td>
<td>- L2 bridge call, SysAdmin try this &amp; that</td>
</tr>
<tr>
<td>2:00pm</td>
<td>- Interrupt Foo app dev, Find sysadmin with access</td>
</tr>
<tr>
<td>2:30pm</td>
<td>- Identify incorrect restarts, Multiple roundtrips to get logs</td>
</tr>
<tr>
<td>5:00pm</td>
<td>- Escalate to SVP, Attempt to get middleware to restart</td>
</tr>
<tr>
<td>6:00pm</td>
<td>- Find middleware engineer, Review docs</td>
</tr>
<tr>
<td>9:30pm</td>
<td>- Need approval, Trial &amp; error</td>
</tr>
<tr>
<td>10:30pm</td>
<td>- Middleware calls restart done</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context</th>
<th>2</th>
<th>10</th>
<th>6</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOC (Bob)</td>
<td>Biz Manager</td>
<td>App Manager</td>
<td>Lead Dev (Karen)</td>
<td>Foo L2</td>
<td>SysAdmin (Lee)</td>
<td>Middleware Mngr. SVP</td>
<td>Chief of Staff</td>
</tr>
</tbody>
</table>

HC: 2 | HC: 9 | HC: 2 | HC: 3 | HC: 4 | HC: 1.5 | HC: 1.5 | HC: 1.5 | HC: 1.5 | HC: 1.5 |
<table>
<thead>
<tr>
<th>Time</th>
<th>HC</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>12:00pm</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2:00pm</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>2:30pm</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>5:00pm</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>6:00pm</td>
<td>1.5</td>
<td>13</td>
</tr>
<tr>
<td>9:30pm</td>
<td>1.5</td>
<td>13</td>
</tr>
<tr>
<td>10:30pm</td>
<td>1.5</td>
<td>13</td>
</tr>
</tbody>
</table>

Don’t underestimate the expense of context switching!
1. Use Lean thinking and analyze like any process
2. Make sure you are measuring the right things
3. “Shift Left” control and decision making
<table>
<thead>
<tr>
<th>Time</th>
<th>Shift-Left</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>-NOC alerts</td>
</tr>
<tr>
<td>12:00pm</td>
<td>+00:30 after alert</td>
</tr>
<tr>
<td></td>
<td>-L2 bridge call</td>
</tr>
<tr>
<td></td>
<td>+02:30 after alert</td>
</tr>
<tr>
<td></td>
<td>-SysAdmin try this &amp; that</td>
</tr>
<tr>
<td></td>
<td>+04:30 after alert</td>
</tr>
<tr>
<td></td>
<td>-Consensus: Foo service</td>
</tr>
<tr>
<td>2:00pm</td>
<td>+04:00 after alert</td>
</tr>
<tr>
<td></td>
<td>-Interrupt Foo app dev</td>
</tr>
<tr>
<td>2:30pm</td>
<td>+05:00 after alert</td>
</tr>
<tr>
<td></td>
<td>-Find sysadmin with access</td>
</tr>
<tr>
<td></td>
<td>-Attempt to get middleware to restart</td>
</tr>
<tr>
<td></td>
<td>-Multiple roundtrips to get logs</td>
</tr>
<tr>
<td>5:00pm</td>
<td>+07:30 after alert</td>
</tr>
<tr>
<td></td>
<td>-Identify incorrect restarts</td>
</tr>
<tr>
<td></td>
<td>-Approve</td>
</tr>
<tr>
<td></td>
<td>-Escalate to SVP</td>
</tr>
<tr>
<td>6:00pm</td>
<td>+08:30 after alert</td>
</tr>
<tr>
<td></td>
<td>-Find middleware engineer</td>
</tr>
<tr>
<td></td>
<td>-VPs try to assess impact</td>
</tr>
<tr>
<td></td>
<td>-Review docs</td>
</tr>
<tr>
<td></td>
<td>-Trial &amp; error</td>
</tr>
<tr>
<td>9:30pm</td>
<td>+12:00 after alert</td>
</tr>
<tr>
<td></td>
<td>-Need API tests per policy</td>
</tr>
<tr>
<td>10:30pm</td>
<td>+13:00 after alert</td>
</tr>
<tr>
<td></td>
<td>-Middleware calls restart done</td>
</tr>
<tr>
<td></td>
<td>-NOC sees green</td>
</tr>
<tr>
<td></td>
<td>-Ticket closed</td>
</tr>
</tbody>
</table>

**Participants:**
- NOC (Bob)
- Biz Manager
- SysAdmin (Steve)
- Lead Dev (Karen)
- Middleware Mngr.
- SVP
- Cust. Engage. (Varsha)
### Shift-Left

<table>
<thead>
<tr>
<th>Time</th>
<th>10:00am</th>
<th>12:00pm</th>
<th>2:00pm</th>
<th>2:30pm</th>
<th>5:00pm</th>
<th>6:00pm</th>
<th>9:30pm</th>
<th>10:30pm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>-NOC alerts</td>
<td>-L2 bridge call</td>
<td>-Interrupt Foo app dev</td>
<td>-Identify incorrect restarts</td>
<td>-Escalate to SVP</td>
<td>-Find middleware engineer</td>
<td>-Need API tests per policy</td>
<td>-Middleware calls restart done</td>
</tr>
<tr>
<td>Participants</td>
<td>NOC (Bob)</td>
<td>Biz Manager</td>
<td>SysAdmin (Steve)</td>
<td>NOC (Bob)</td>
<td>Biz Manager</td>
<td>App Manager</td>
<td>Lead Dev (Karen)</td>
<td>NOC (Bob)</td>
</tr>
<tr>
<td>Location</td>
<td>Solve here or here</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Solve here or here**

- NOC alerts
- L2 bridge call
- Biz Manager phone call
- SysAdmin try this & that
- Escalate to L2
- Consensus: Foo service
- Interrupt Foo app dev
- Find sysadmin with access
- Multiple roundtrips to get logs
- Identify incorrect restarts
- Attempt to get middleware to restart
- Escalate to SVP
- VPs try to assess impact
- Approve
- Find middleware engineer
- Review docs
- Trial & error
- Pass tests
- Find tester
- Need API tests per policy
- Middleware calls restart done
- NOC sees green
- Ticket closed

---

**Timeline**

- **10:00am**
  - NOC alerts
  - Biz Manager phone call
- **12:00pm**
  - L2 bridge call
  - SysAdmin try this & that
- **2:00pm**
  - Escalate to L2
  - L2 bridge call
- **2:30pm**
  - Consensus: Foo service
  - Interrupt Foo app dev
  - Find sysadmin with access
  - Multiple roundtrips to get logs
- **5:00pm**
  - Identify incorrect restarts
  - Attempt to get middleware to restart
- **6:00pm**
  - Escalate to SVP
  - VPs try to assess impact
  - Approve
- **9:30pm**
  - Find middleware engineer
  - Review docs
  - Trial & error
  - Need API tests per policy
- **10:30pm**
  - Middleware calls restart done
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00am</td>
<td>1. NOC alerts</td>
</tr>
<tr>
<td></td>
<td>- Biz Manager phone call</td>
</tr>
<tr>
<td></td>
<td>- Escalate to L2</td>
</tr>
<tr>
<td></td>
<td>- NOC (Bob) Biz Manager</td>
</tr>
<tr>
<td></td>
<td>- SysAdmin (Steve) 7 x L2</td>
</tr>
<tr>
<td></td>
<td>- Solve here or here</td>
</tr>
</tbody>
</table>

| 12:00pm | 2. L2 bridge call |
|         | - Biz Manager |
|         | - SysAdmin try this & that |
|         | - Multiple roundtrips to get logs |
|         | - NOT HERE |

| 2:00pm | 3. Interrupt Foo app dev |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 2:30pm | 4. Identify incorrect restarts |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 2:30pm | 5. Attempt to get middleware to restart |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 5:00pm | 6. Escalate to SVP |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 5:00pm | 7. Need API tests per policy |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 5:00pm | 8. Need approval |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 9. Find middleware engineer |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 10. Find sysadmin with access |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 11. Attempt to get middleware to restart |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 12. Identify incorrect restarts |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 13. Approve |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 14. Review docs |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 15. Trial & error |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 6:00pm | 16. Pass tests |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 9:30pm | 17. Middleware calls restart done |
|        | - NOC (Bob) Biz Manager |
|        | - App Manager |
|        | - Lead Dev (Karen) |
|        | - Foo L2 |
|        | - SysAdmin (Lee) |
|        | - middleware engineer |

| 10:30pm | 18. Ticket closed |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | -Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 19. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 20. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 21. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 22. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 23. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 24. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 25. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 26. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |

| 10:30pm | 27. Cust. Engage. (Varsha) |
|         | - NOC (Bob) Biz Manager |
|         | - App Manager |
|         | - Lead Dev (Karen) |
|         | - Foo L2 |
|         | - SysAdmin (Lee) |
|         | - middleware engineer |
Empower those closest to the issue
Empower those closest to the issue

Push the ability to take action this direction
Empower those closest to the issue

Push the ability to take action this direction

But what gets in the way?
Empower those closest to the issue

Push the ability to take action this direction

But what gets in the way?

Silos
1. Use Lean thinking and analyze like any process
2. Make sure you are measuring the right things
3. “Shift Left” control and decision making
4. Get rid of silos
Silos ruin everything
Silos ruin everything

I need X

Silo A
- Backlog
- Context
- Tools
- Priorities

Requests for X

I do X

Silo B
- Backlog
- Context
- Tools
- Priorities
Ticket-Driven Request Queues Are Best Indicator of Silos

Team A
(Dev)

Ticket System

Team B
(Ops)
Ticket-Driven Request Queues Are Best Indicator of Silos

Silo Builder
Ticket-Driven Request Queues Are Best Indicator of Silos

Silo Builder

Snowflake Maker
Popular: Replace Silos with Cross Functional Team
Popular: Replace Silos with Cross Functional Team

Cross-Functional Teams

Cross-Functional Teams

Cross-Functional Teams
Popular: Replace Silos with Cross Functional Team

Planning  |  Dev/Test  |  Release  |  Operate
---|---|---|---
Cross-Functional Teams
Cross-Functional Teams
Cross-Functional Teams
Popular: Replace Silos with Cross Functional Team

Cross-Functional Teams

Cross-Functional Teams

Cross-Functional Teams

DBAs  Environments  Network  Security  NOC
Popular: Replace Silos with Cross Functional Team
1. Use Lean thinking and analyze like any process
2. Make sure you are measuring the right things
3. “Shift Left” control and decision making
4. Get rid of silos
5. Establish Operations as a Service
Get rid of as many ticket-driven request queues as possible

Silo Builder
Get rid of as many ticket-driven request queues as possible

Silo Builder

Snowflake Maker
... replace with Operations as a Service design pattern

Team A (Dev)

Define Procedures
Execute On Demand

Operations as a Service

Vet Procedures
Define Policies
Execute On Demand

Team B (Ops)

Ticket System

Actual Exceptions
Changes how your organization thinks about automated procedures…
Automated procedures are comprised of three parts

- **Define**
  - Definition of the automated procedure

- **Execute**
  - Execution of the automated procedure

- **Govern**
  - Governance of the automated procedure
Automated procedures are comprised of three parts

- **Define**
  - Definition of the automated procedure

- **Execute**
  - Execution of the automated procedure

- **Govern**
  - Governance of the automated procedure
    - (security, oversight, compliance, etc.)
Traditional Ops Silo

“Consumers of Ops”  
(Dev, QA, Release, NOC, Security, etc.)

Define
Execute
Govern

Ops
Rigid Self-Service

"Consumers of Ops"
(Dev, QA, Release, NOC, Security, etc.)
Rigid Self-Service (limited)

“Consumers of Ops” (Dev, QA, Release, NOC, Security, etc.)

Define

Execute

Govern

Ops
High-Velocity Handoffs

Define

Govern

Execute

Ops

“Consumers of Ops”
(Dev, QA, Release, NOC, Security, etc.)
Self-Service Operations

“Consumers of Ops”  
(Dev, QA, Release, NOC, Security, etc.)
Self-Service Operations

Define

Govern

"Consumers of Ops"
(Dev, QA, Release, NOC, Security, etc.)

Govern

Execute

Ops

Govern
Operations as a Service

Team A (Dev)
- Define Procedures
- Execute On Demand

Team B (Ops)
- Define Procedures
- Execute On Demand

Operations as a Service
Operations as a Service

Move definition, execution, and governance to where you get the most effective use of labor and best flow of work.
Rundeck: Open Source Platform For Operations as a Service

Create workflows • Define ACL policies • Execute workflows

Web GUI • API • CLI

Orchestration & Scheduling of Workflows
Collect and Process Output

Infrastructure details and state from multiple sources

Corp Directory
Authentication and roles

Tickets, work status, approvals

ITSM

CMDB
Monitor.
Metrics
Cloud
Config. Man.

Scripts • APIs • Tools • Cloud • VMs • Containers
Common implementation pattern for Operations as a Service…
Step 1: Establish a Secure Ops Hub

**Operations as a Service**

- **Ops Procedures**
  - "Status"
  - "Firewall Change"
  - "Restart"

- **Inventory and Health**
  - Infrastructure view
  - Service health
  - System metrics

**RUNDECK + Monitoring Tools**

- **Secrets**
- **Identity**
- **Audit Logs**

**Security and Ops** manages access, configuration, and compliance

**Engineers** get visibility and controlled self-service

**Ops Support** use for remediation procedures

**Execute**
Step 2: Establish a SDLC for Ops Procedures

Operations as a Service

- **Ops Procedures**
  - “Status”
  - “Firewall Change”
  - “Restart”

- **Inventory and Health**
  - Infrastructure view
  - Service health
  - System metrics

**Rundeck** + Monitoring Tools

- **Secrets**
- **Identity**
- **Audit Logs**

- **Execute**

- **Product Engineers**
  - Produce automated procedures and health checks.

- **Security and Ops**
  - Manage access, configuration, and compliance

- **Engineers**
  - Get visibility and controlled self-service

- **Ops Support**
  - Use for remediation procedures

Automated Procedures and Health Checks

- **Source Code Repo**
  - Git

- **Code review**

- **RISKY**

- **FIX**

- **if** ($state==wait)
  - then
  - kill -9 $PID
  - fi

- **FI**

- **Fix**

- **Product Engineers**
  - Produce automated procedures and health checks.

- **Security and Ops**
  - Manage access, configuration, and compliance

- **Engineers**
  - Get visibility and controlled self-service
Step 3: Connect with Enterprise Management Systems

- **Engineers** get visibility and controlled self-service.
- **Ops Support** use for remediation procedures.
- **Ops Support** get visibility and audit trail updated by support tools.

**Operations as a Service**

- **Ops Procedures**
  - “Status”
  - “Firewall Change”
  - “Restart”
- **Inventory and Health**
  - Infrastructure view
  - Service health
  - System metrics

**Source Code Repo**

- **Product Engineers** produce automated procedures and health checks.

**Security and Ops** manage access, configuration, and compliance.

**Secrets**

**Identity**

**Audit Logs**

**RISKY**

**FIX**

**Automated Procedures and Health Checks**

**Code review**

**Execute**

**Service Desk**

**Service Ticket**

**Software Supply Chain**

**Customers**

**Ops integrate with artifact flow**
Step 4: Make Compliance Really Happy

Who reviewed it?

Who ran it? When? Where?

Approval trail?

Who created the procedure?

Who created the policy?

if (($state==wait)) then
   kill -9 $PID
fi

Code review

Change

Source Code Repo

git

Operations as a Service

Ops Procedures
- "Status"
- "Firewall Change"
- "Restart"

Inventory and Health
- Infrastructure view
- Service health
- System metrics

Secrets

Identity

Audit Logs

allow
deny

Product Engineers produce automated procedures and health checks.

Security and Ops manages access, configuration, and compliance

Fix

RISKY

Automated Procedures and Health Checks

Operations as a Service

Engineers get visibility and controlled self-service

Ops Support use for remediation procedures

Ops Support get visibility and audit trail updated by support tools

Execute

Service Desk

Service Ticket

Software Supply Chain

Nexus

Ops integrate with artifact flow

Customers

Who reviewed it?

Who ran it? When? Where?

Approval trail?

Who created the procedure?

Who created the policy?

Code review

Change

Source Code Repo

git

Operations as a Service

Ops Procedures
- "Status"
- "Firewall Change"
- "Restart"

Inventory and Health
- Infrastructure view
- Service health
- System metrics

Secrets

Identity

Audit Logs

allow
deny

Product Engineers produce automated procedures and health checks.

Security and Ops manages access, configuration, and compliance

Fix

RISKY

Automated Procedures and Health Checks

Operations as a Service

Engineers get visibility and controlled self-service

Ops Support use for remediation procedures

Ops Support get visibility and audit trail updated by support tools

Execute

Service Desk

Service Ticket

Software Supply Chain

Nexus

Ops integrate with artifact flow

Customers

Who reviewed it?

Who ran it? When? Where?

Approval trail?

Who created the procedure?

Who created the policy?
1. Use Lean thinking and analyze like any process
2. Make sure you are measuring the right things
3. “Shift Left” control and decision making
4. Get rid of silos
5. Establish Operations as a Service
6. Encourage Developers to think “Operable First”
Encourage Developers to think “Operable First”
Encourage Developers to think “Operable First”

☑️ Operations is the business (unless you literally sell packaged software)
Encourage Developers to think “Operable First”

- Operations is the business (unless you literally sell packaged software)
- Deployability, configurability, monitoring are service features
Encourage Developers to think “Operable First”

- Operations is the business (unless you literally sell packaged software)
- Deployability, configurability, monitoring are service features
- Build configurability into the service, don’t externalize it
Encourage Developers to think “Operable First”

- Operations is the business (unless you literally sell packaged software)
- Deployability, configurability, monitoring are service features
- Build configurability into the service, don’t externalize it
- Demand “prod-like” environments everywhere
Encourage Developers to think “Operable First”

- Operations is the business (unless you literally sell packaged software)
- Deployability, configurability, monitoring are service features
- Build configurability into the service, don’t externalize it
- Demand “prod-like” environments everywhere
- Make any handoff between teams “verification-driven”
Encourage Developers to think “Operable First”

- Operations is the business (unless you literally sell packaged software)
- Deployability, configurability, monitoring are service features
- Build configurability into the service, don’t externalize it
- Demand “prod-like” environments everywhere
- Make any handoff between teams “verification-driven”
- Create immutable versioned artifacts and use standard packaging
Encourage Developers to think “Operable First”

- Operations is the business (unless you literally sell packaged software)
- Deployability, configurability, monitoring are service features
- Build configurability into the service, don’t externalize it
- Demand “prod-like” environments everywhere
- Make any handoff between teams “verification-driven”
- Create immutable versioned artifacts and use standard packaging
- Integration tests over unit tests
Let’s look at a company improving it’s incident response capability…
Before Support at the Edge

TMOL WebOps Incidents

Average MTTR
47 Minutes
Before Support at the Edge

TMOL WebOps Incidents

Average MTTR

47 Minutes
90% Reduction in MTTR
50% Reduction in escalations
55% Reduction of overall support costs
1. New org, support, and escalation model
1. New org, support, and escalation model
1. New org, support, and escalation model

- **1°**: EMT
- **2°**: ER
- **3°**: Trauma Surgeon
- **4°**: Specialist Surgeon

**TOC (NOC)**

**SRE**

**Production Eng. Scrum Teams**

**Data Services Platform Eng. Global Network**

- EMT > 15 min
- ER > 30 min
- Trauma Surgeon > 60 min
- Specialist Surgeon

- Escalate
1. New org, support, and escalation model

2. **Key**: Push the ability to take action closest to the problem
1. New org, support, and escalation model

- TOC (NOC)
- SRE
- Production Eng. Scrum Teams
- Data Services Platform Eng. Global Network

```
> 15 min -> > 30 min -> > 60 min
```

2. **Key**: Push the ability to take action closest to the problem

3. Longterm investment in operability
   (deployment, configuration, monitoring, automated runbooks)
“Support at the Edge”

Sources: https://www.youtube.com/watch?v=_hr4KiB19bQ
http://rundeck.org/stories/mark_maun.html
ticketmaster® “Support at the Edge”

• Automated Ops procedures written/vetted by the delivery teams

Sources: https://www.youtube.com/watch?v=_hr4KiB19bQ
http://rundeck.org/stories/mark_maun.html
“Support at the Edge”

- Automated Ops procedures written/vetted by the delivery teams
- Ops remained in full control of what can run and security policy

Sources: https://www.youtube.com/watch?v=_hr4KiB19bQ
http://rundeck.org/stories/mark_maun.html
“Support at the Edge”

- Automated Ops procedures written/vetted by the delivery teams
- Ops remained in full control of what can run and security policy
- Empowered NOC and other support teams with self-service ops tasks

Sources: https://www.youtube.com/watch?v=_hr4KiB19bQ
http://rundeck.org/stories/mark_maun.html
ticketmaster® “Support at the Edge”

- Automated Ops procedures written/vetted by the delivery teams
- Ops remained in full control of what can run and security policy
- Empowered NOC and other support teams with self-service ops tasks
- Empowered developers with limited self-service operations

Sources: https://www.youtube.com/watch?v=_hr4KiB19bQ
http://rundeck.org/stories/mark_maun.html
“Support at the Edge”

- Automated Ops procedures written/vetted by the delivery teams
- Ops remained in full control of what can run and security policy
- Empowered NOC and other support teams with self-service ops tasks
- Empowered developers with limited self-service operations
- Combined with new incident response and escalation model

Sources: https://www.youtube.com/watch?v=_hr4KiB19bQ
http://rundeck.org/stories/mark_maun.html
Recap!

1. Use Lean thinking and analyze like any process
2. Make sure you are measuring the right things
3. “Shift Left” control and decision making
4. Get rid of silos
5. Establish Operations as a Service
6. Encourage Developers to think “Operable First”
Let’s talk…

damon@rundeck.com

@damonedwards